ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Providing New Architecture for Self-Compatible Software

VahidRezaei
Dept. of Software, College of Computer, Birjand Branch, Islamic Azad University, Birjand, Iran.

Abstract
Today's application programs, must challenge with variable needs which itself cause increasing dynamic, complication in resources control and autonomy in present environments. Feature of autonomy and compatibility against user's needs, introduce this method as one of applicable methods of software production. For foundation this system it must be noted that system must remedy itself without human involvement, replaces components and totally makes changes to environment and fits system with considering new conditions. Self-compatibility, is interesting trend for response to dynamic needs of software. With attention to given definitions in software engineering field, we can say that self-compatible software, assesses their treatment and when it is defined that it does not work correctly or could work with more efficiency, it changes its treatment. In provided architecture for resolution problems of distributed systems, architecture will develop self-compatibility for distributed systems and by use of PSO algorithm will optimize it enabling different distribution systems to use different resources with attention to their needs and satisfy needs and difficulties.

I. Introduction

Self-compatible software entered quickly in new challenge of complicated calculations and got importance. For achievement of compatibility, software must be implemented designed by suitable methods and strategies. Previously, most of compatibilities were done by developing answers at level of software architecture[1].

By increasing demand for software production and application programs, software must be designed so that having responsive to future needs of society. Therefore, for improvement of efficiency in thise software, methods called self-remedy, self-protective, self-compatible, self -improvement and self-organization have been introduced. By attention to continuous implementation of system and different factors which cause involvement in system's work, so the systems were failed, therefore, system must be foundation so that to protect stable situation and position of system. Self-compatible software and self- calculations give methods for this activity. Researches in this field led to inter discipline relations and involvement of ranges of disciplines related to artificial intelligence for controlling theories and software engineering. For achievement of these aims, answers related to self-compatible treatment of engineers were mostly at different architecture levels of software which include medium wares and design based on components. According to architecture methods for dynamic compatibility comprehensively, many researches have been performed. Components of self-compatibility are formation units of system which monitor resources and give results or components to users. These components are related with one or many managed resources [3].

There are diversity researches in field of compatible approaches in software systems. Principally, software which have trend of open ring, by a feedback ring will become closed ring system. This compatibility may be achieved through feedback forward like control work volume and this feedback ring, will have control on events inside software. Self-compatible softwareare developed with aim of setting different products and or features in response to its variations and in frame work of software system. Among main aims of self-compatible software systems growth, there are management complications, strength against non-predicted conditions, changing priorities and government policies and changing situations. Today's application programs, must challenge with changing needs and restrictions, which itself cause increasing dynamics, complication in resources control and autonomy in present environments. Feature of autonomy and compatibility against user changing needs, introduces this method as one of most applicable one in software production. For foundation this system, it must be noted that system must remedy itself without human involvement, replaces components and totally make changes in environment and make system compatible with new conditions. It is clearly seen that self-compatible software, have significant role in future software systems and information technology [4].

In this study, an architecture is presented to use self-compatible architecture but in distribution form, and raised for problems of different systems. In proposed architecture, it is assumed that architecture layers are designed as different types, because it can response to different systems and non-homogenous needs.

II. Study method of Rainbow framework

Aim of Rainbow framework is general structure of self-compatibility so that different parts of a system which are gathered and form a family of system, enabling to use repeatedly. This architecture includes layers: infrastructure layer of particular system, architecture layer and translation layer. Architecture model studies restriction events periodically to remove restrictions and makes compatibility. Compatibility motor will determine needed operation conditions. Translation layer responsibility is relation between system layer and architecture layer [5].

III. Autonomic Management Engine (AME)

Aim of autonomic management is presenting a general architecture, with complete tools (foolkit) for addition a self-compatible method with use of ring control to existing programs. Events show existence of first class and includes all types of corresponding data such as reporting components and damaged components and events conditions by application programs are produced. AME architecture around message passage, distributed event implementation event in occurrence and distribution of possibilities for this event and performance manager have function of management and analysis, gathering each component and class analysis. Each way includes body event of resource model which this model includes special system which has knowledge of compatibility for doing a special application (fig. 1) [6].

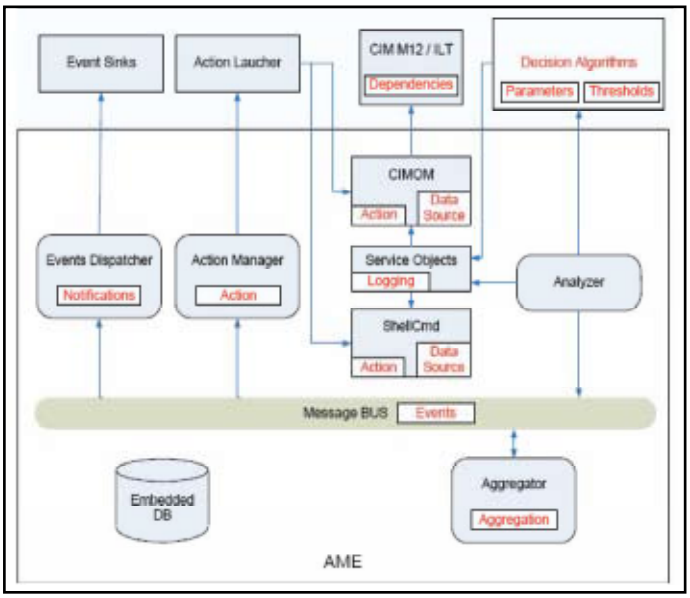

Fig. 1: Architecture of Autonomic Management Engine

IV. Distributed systems

By passing time and recent decades, systems work distributed and on clouds which has many advantages. Accessibility of distributed system which facilitates user works it is sufficient with a communication bed such as internet we have access to distributed system with cloud. Perhaps, the system which we work on has not enough hardware and software possibilities. By method of communication with distributed system there is no need for all of hardware and software facilities and it has importance for users. Users easily receive their needed services by connecting to distributed system (figure 2) [7].

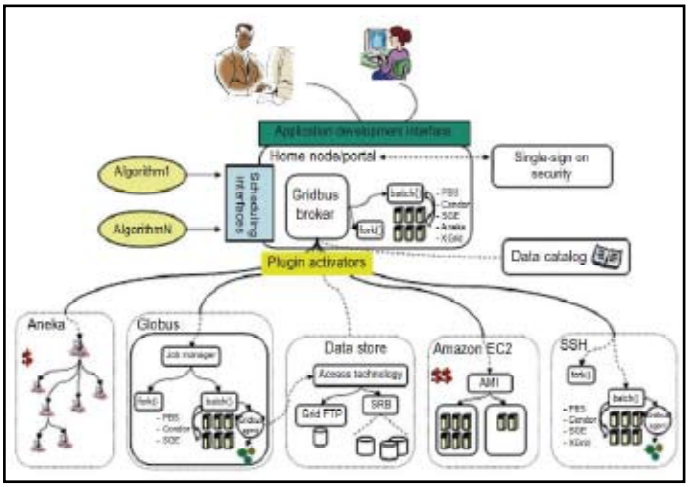

Fig. 2 : Distributed system

V. Optimizing algorithm (PSO)

One of important algorithms which is used for optimizing problems is PSO algorithm. This algorithm by using social models and existing relations will simulate performance of algorithms. In performing PSO algorithm, many parameters such as number of particles (usually 10-50) are involved which setting up them suitably will affect algorithm performance intensively. In this algorithm, each answer to problem is modeled as a particle which has a value and also proportional rate. Each particle in each stage, remembers the position which has best result (best individual position of each particle). Information particles will exchange with each other about position which they are. All particles select a direction for movement and after movement one stage of algorithm is finished. These stages are repeated many times until desired answer is obtained [8].

PSO has many advantages than other methods of optimizing, among them: is a zero order method and no need to heavy mathematical operations such as gradation, is based on population (using distributed calculations); acceptable tthere is no selection operation. It means that no particle (answers) is deleted and only each particle value is changed; does not use generation survival principle; there is no combination of answers (cross over); in PSO also mutation is available somehow [9].

VI. Distributed Autonomic Management Engine

Presented architecture has been described with attention to importance of distributed systems and automatic self-compatibility system performance. Raised architecture is precisely based on Autonomic Management System, but due to study this method in distributed environment, defect systems and needs to systems for continuation work implementation need a management and method which have access to best possible status for work implementation. This architecture has developed work of self-compatible systems as distributed, so that users can access to resources in distributed form and different users system in case of any problem in system will have access to nearest resources very quickly. Therefore, cost spent for implementation self-compatibility architecture is divided through more systems and is more economic. It is noted that such an architecture has application opacity and is hidden in users view and users will not be noticed about problem in system and its resolution. This is function of self-compatible architecture in distributed systems [10].

For management and accessing defect system to best possible status for going on working and selection best system, this part of selection in this architecture was done based on PSO algorithm (figure 3), because, PSO algorithm in case of congestion and particles bulk (parts and systems) will do best selection.

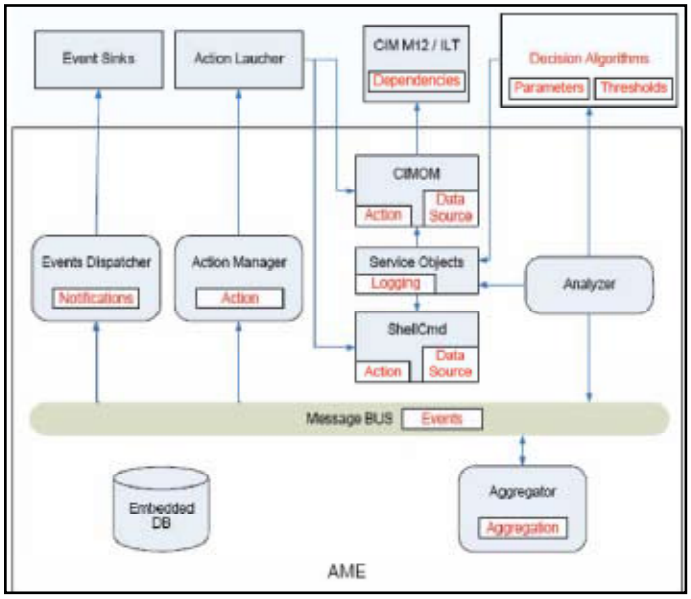

Fig. 3 : Distributed Autonomic Management Engine Architecture

VII. Implementation method

In this program, 4 types of architecture layers have been considered to enable problem to be resolved for different systems. In each

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

architecture series there are many types of resources to debugging each system according homogeneity or architecture type and needing its resources and by access to needed resource continue its working process. So, x axis is healthy resources and y function is allocation resources to defect system with attention to needs and has been considered as a maximum needs.

Each system finds its resource problems as optimum with PSO algorithm and all of its systems needs will be accessed by needed resources in optimized form. In fact, in diagram, 4 types of different architecture have been given for self-compatibility which everyone has been considered as a function which does specific job and investigates system problems with specific strategy and architecture, automatically. In fact, here, 4 different types of model have been considered to respond users with different systems for homogeneity with self- compatibility architecture, which with attention to environment magnitude has been distributed and developed which each system requires healthy resources (x) and defect system (y) (figure 4) [11].

Self-compatibility architecture method has been optimized by PSO algorithm (figure 5). Systems achieve their needed resources in best manner. By using PSO algorithm, systems search the path and reach to the best answer, so, system will access to resources. PSO algorithm helps working process of automatic management architecture in distributed systems, because it studies best possible status for removing problem from a distributed system [12].

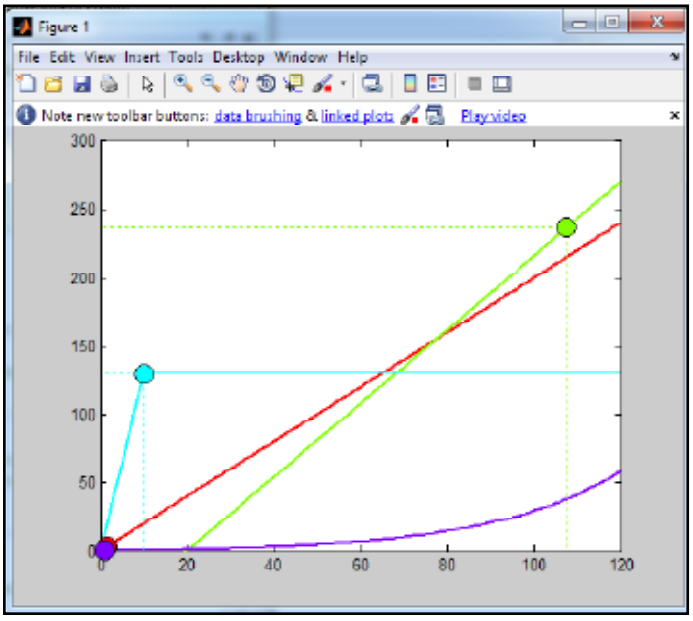

Fig. 4 : Diagram for self-compatible distributed architecture for 4 types of architecture layers with using PSO algorithm

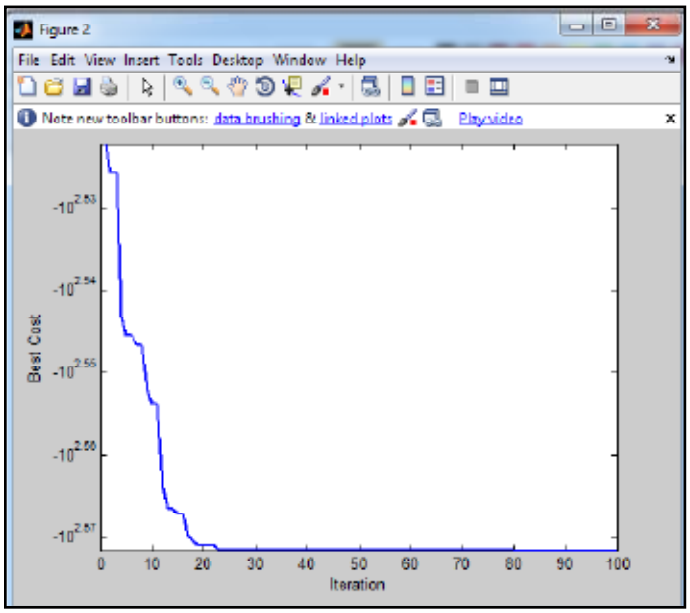

Fig. 5 : Diagram for optimized allocation of resources to systems with PSO algorithm

VIII. Conclusion

Systems which have got problems, need removing of problems which many types of architectures have been presented for resolution of these problems. In old architectures, there was need for recognition problem by human and removing it which was very time consuming and costly. But, in self-compatibility architecture, system itself notices system defect and resolves the problem. Considering the importance of distributed systems in architecture given for resolution problems of distributed systems, self-compatible architecture has been developed for distributed systems and optimized by PSO algorithm, enabling different distributed systems to use different resources according to their needs and removing needs and problems.

References and Resources:

[1] *Anderson, J., de Lemos, R., Malek, S., Weyns,D.: Modeling Dimensions of Self–Adaptive Software Systems. In: Cheng, B.H.C., de Lemos, R., Gies, H., Inverardi, P., Magee, J. (eds.) Self-Adaptive Systems. LNCS, vol. 5525, pp. 27–47. Springer, Heidelberg (2009)*

[2] *B. H. C. Cheng and J. M. Atlee.Research directions in requirements engineering. In FOSE '07: 2007 Future of Software Engineering, pages 285{303. IEEE Computer Society, Minneapolis, MN, USA, May 2007*

[3] *Dongmin, Z., Dachao, H. and Yuchun, X., 'A Framework for Ontology-based Product Design Knowledge Management', IEEE Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol.4, pp.1751 – 1755, 2010.*

[4] *E. Manoel, M. J. Nielsen, A. Salahshour, S. Sampath, and S. Sudarshanan.Problem Determination Using Self-Managing Autonomic Technology.IBM redbooks, June 2005.*

[5] *Strassner, J.; Samudrala, S.; Cox, G.; Liu, Y.; Jiang, M.; Zhang, J.; Meer, S.; Foghl´u, M.; Donnelly, W. The design of a new context-aware policy model for autonomic networking. In ICAC '08: Proceedings of the 2008 International Conference on Autonomic Computing, pages 119–128, Washington, DC, USA, 2008. IEEE Computer Society.*

[6] *Wang, Y., Guo, J., Hu, T. and Wang, Y., 'An Ontology-Based*

Framework for Building Adaptable Knowledge Management Systems', Springer-Verlag Berlin Heidelberg, pp. 655–660, 2007.

[7] Weyns. D., Haesevoets. R., Helleboogh. A., Holvoet. T., and Joosen. W. The MACODO Middleware for Context-Driven Dynamic Agent Organzations. ACM Trans- action on Autonomous and Adaptive Systems, 2010.

[8] Zhang, J., Zhao, W., Xie, G. and Chen, H., 'Ontology- Based Knowledge Management System and Application', Elsevier Procedia Engineering Advanced in Control Engineering and Information Science, Vol.15, pp.1021-1029, 2011.

[9] Zhang, L., Xia, Sh., Zhou, Y. and Xia, Z., 'User Defined Ontology Change and its Optimization', IEEE Chinese Control and Decision Conference, pp. 3586 – 3590, 2008.

[10] Zhang, R. and Guo, L., 'Knowledge Management Based on Ontology Modeling in Collaborative Learning Environment', IEEE Computer Society International Conference on Computer Science and Software Engineering, 2008.

[11] Zhou, M. and Tao, J., 'A Framework for Ontology-Based Knowledge Management', IEEE International Conference on Business Management and Electronic Information (BMEI), Vol.4, pp.428 – 431, 2011.