Survey on SOA Fault Tolerance Techniques

Anushka

M.Tech (IT)Student, Dept. of ISE, R.V.College of Engineering, Bangalore,India

Abstract: The collection of discrete software modules forming software architecture design pattern to deliver services. These services collectively provide the complete functionality of a large software application. SOA allows performing the operations of a large number of computers that are connected over a network with ease. SOA exhibits features loose coupling, black box, dynamism, etc. Sometimes, SOA faces problem in delivering messages between different services and faults in the SOA components, etc. These faults are diagnosed with various methods such as actuators, different service routing methods, MAPE, etc.

Key Words: SOA, Fault Management, MAPE, Double Redundancy, Service Routing.

I. INTRODUCTION

Service-Oriented Architecture (SOA) is an effective model for developing applications. SOA discloses features which are not commonly found in conventional computing paradigms [7]. The features of services, loose coupling, dynamism, blackbox, and heterogeneity, make service management more challenging than conventional system management. Consequently, SOA service management face problems of increased management effort, decreased management effectiveness, and irresolvable service faults. There are many resolutions to these ill operations and this paper will be discussing some of them [1]. This paper will discuss the faults, problems that has been identified by authors occurring in SOA and their different technique used to resolve them. Authors of “An Engineering Process for Autonomous Fault Management in Service Oriented Systems” promises an approach to manage faults in SOA is to apply the principle of autonomous computing (AC) [12], on service management, where AC is known as an efficient approach to manage systems in autonomous manner with minimum human intervention. To apply AC to managing faults in SOA, 1) the whole life-cycle model and 2) instructions for carrying out each phase are both required. 3) Also, a key phase which requires a mathematically sound foundation in this life-cycle is to diagnose the given symptoms/faults and determine their underlying causes. Moreover, it is required that the process supports to make the autonomous management agent/system more intelligent, by 4) learning new facts and rules from the experiences of applying each cycle. This is an essential requirement to be autonomous since human intervention is not an integral part of AC and can be called as engineering process. Hence, in engineering process it evolves following steps. First, define a whole life-cycle process for managing faults in SOA as well as the essential artifacts of each phase. Then, present a set of instructions for five phases; Preparation Phase, Monitoring Phase, Diagnosis Phase, Healing Phase, and Learning Phase.

Author of “Double Redundant Fault-Tolerance Service Routing Model in ESB” [2],[6] believes that Enterprise Service Bus (ESB) is a backbone for service integration. Usually, it is regarded ESB as the kernel of SOA, and it can manage mass services for different purposes. An ESB [19] acts as a standards-based integration platform which syndicates messaging, web services and data transformation in a highly distributed environment. The Author concentrates more on how to provide a reliable service routing so that despite of any faults in routing the messages has to be reached to the destination. Hence, they introduced a double service routing method wherein there will be replica of each service routing so that if one service routing fails to deliver the message then there will be another to deliver it.

Authors of “A Practical Framework of Realizing Actuators for Autonomous Fault Management in SOA” [3] aim to resolve the faults that occur in the components of SOA system. They came up with an approach applying the principles of autonomic computing (AC) [10], [12], which will manage the services in consistent and dynamic manner. The first two phases of MAPE [3],[13] are to monitor target systems and diagnose faults to determine underlying cause. The other two phases of it are to plan healing/actuation methods and to execute them. They introduced Actuators the software modules which will act as a healing mechanism to the faulty components based on their symptoms. They presented a practical framework to design actuators which can be invoked autonomously in SOA.
environment, by considering the relationship among fault, source, and actuator.

II. LITERATURE REVIEW

In “An Engineering Process for Autonomous Fault Management In Service Oriented Systems” Service-Oriented Computing exposes features such; loose coupling, black box, dynamism and heterogeneity [1][7]. Thus, service-oriented systems often face problems of augmented cost-effort, reduced effectiveness, and irresolvable service faults. Du Wan Cheun and Soo Dong Kim exhibit a method to manage faults in autonomous manner [11]. It includes a complete life-cycle process to manage faults in SOA and the important artifacts of each stage, which carries out some instructions. Each stage consists of set of instructions - Preparation Phase, Monitoring Phase, Diagnosis Phase, Healing Phase, and Learning Phase [8]. They suggest methods for diagnosing faults which is devised with two inference mechanisms, i.e. Model-Based Reasoning (MBR) [15][17], - Probability-based reasoning (PBR) and reasoning knowledgebase updates by learning mechanism in detailed. The process is carried out on current service-oriented architecture standards.

In “Double Redundant Fault-Tolerance Service Routing Model in ESB the development of the Service Oriented Architecture (SOA), Enterprise Service Bus (ESB)]2 [6] is becoming more important in the management of mass services. The main feature is to deliver the messages between different services by using service of routing. Presently static configuration service routing is being used widely. If any one message delivering service fails, the system could not identify this fault. Bin Wu, Shijun Liu and Wei Cui is resolve this problem, they present a double redundant fault tolerant service routing model. They propose an algorithm to achieve the double redundant fault tolerant mechanism. If the original service fails, an alternate replica service will be activated that has the same function and it will return the response message automatically [8]. The service requester receives the response message evidently without any knowledge that where it comes from. Thus, the response message service will be delivered even there be any problem ensuring the reliability. The service that has failed to perform its job will be recorded by service management. Also, the performance of double redundant fault tolerant service routing model are evaluated. By introducing double redundant fault tolerance by implementing dynamic routing, and ensure the reliability of messaging in SOA.

In “A Practical Framework of Realizing Actuators for Autonomous Fault Management in SOA” the faults in SOA can be diagnosed by applying the principles of autonomic computing (AC)[9], [10][12], of which process is specified in Monitor, Analyze, Plan, Execute (MAPE). MAPE comprises of 2 stage[3], firstly to monitor target systems and diagnose faults to determine underlying cause. Secondly it plans the healing or actuation methods and to execute actuators applying on faults. Hyun Jung La and Soo Dong Kim, present a practical framework to design actuators which can be invoked autonomously. By seeing the relationships between fault, cause, and actuator, they derive the abstract and concrete actuators [9]. For these actuators, they derive algorithms which can be implemented in practice on faults. Thus, the faults can be analyzed by implementing respective actuators.

In “SOAR: An Extended Model-Based Reasoning for Diagnosing Faults in Service-Oriented Architecture”. Service-Oriented Architecture (SOA) is a cost effective approach to build enterprise applications [4]. SOA reveals non-conventional challenges such as fault diagnosis at runtime is challenging due to the SOA features. Model-Based Reasoning (MBR) [17] is a formal approach to analyze faults, which is built on predicate calculus and term resolution. Soo Dong Kim and Soo Ho Chang present SOAR [9] (Service-Oriented Abductive Reasoning) which covers the basic MBR [16], [18] to diagnose faults in copious SOA components. SOAR delivers an enhanced inference capability by comparing the monitored state based and QoS-based reasoning in addition to the basic setting or observation-based reasoning [13]. They proposed a method to formally represent system description i.e. system component description, components normal behaviour [9],[10], [12], fault model and observations, and reasoning methods to analyze faults and to identify their causes.

In “The Sender Released Pattern: An SOA design pattern for inter-service message Exchange” it describes modelling a complex interactions and integrations between distributed and heterogeneous applications Service-oriented architectures are widely being used [5]. But, these architectures possess some failures (e.g. reliability, availability, and performance problems). Design patterns, can be used as tested solutions to specific problems. The “Reliable
Messaging” pattern is proposed by the SOA design pattern community, its reliable system is subjected to a great number of failures that can crash the dependable functionality [14]. Imen TOUNSI, Mohamed Hadj Kacem, proposed a service based approach for dynamic management of service-oriented architectures using SOA design patterns [20]. They introduced a new pattern that extends the “Reliable Messaging” pattern in order to resolve its fault and also introduced message-oriented SOA design patterns model with the SoaML [21], standard language through “Sender Released” pattern.

III. CRITICAL EVALUATION

<table>
<thead>
<tr>
<th>Literature Reference</th>
<th>Problem discussed</th>
<th>Technique Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Service Faults</td>
<td>Two inference mechanisms; MBR (model based reasoning) and PBR (Probability based reasoning).</td>
</tr>
<tr>
<td>2</td>
<td>Service routing</td>
<td>A double redundant fault tolerant service routing model</td>
</tr>
<tr>
<td>3</td>
<td>Faults in the SOA components</td>
<td>Deriving the abstract and concrete actuators.</td>
</tr>
<tr>
<td>4</td>
<td>Diagnose fault at runtime in SOA components</td>
<td>SOAR (Service-Oriented Abductive Reasoning) which covers the basic MBR</td>
</tr>
<tr>
<td>5</td>
<td>Faults in the interactions and integrations between distributed and heterogeneous applications Service-oriented architectures</td>
<td>Design patterns : “Sender Released” pattern, the modeling of the “Sender Released” pattern with the help of SoaML language.</td>
</tr>
</tbody>
</table>

IV CONCLUSION

This paper discusses about the various SOA faults, techniques to identify, analyse and resolve these faults.

REFERENCES


