SOA Adaptation using Fuzzy Theory in Context-Aware Mobile Computing Model

1Sahba Paktinat, 2Saeed Arezi,
3Abdullah Bahmani and 4Afshin Salajegheh
Computer Engineering Department, Islamic Azad University, South Tehran Branch, Iran. Sahbap67@gmail.com
Computer Engineering Department, Islamic Azad University, Zanjan Branch, Iran. saeedarezi@gmail.com
Computer Science Department, Payame Noor University, Zanjan Branch, Iran. abahmani@gmail.com
Computer Engineering Department, Islamic Azad University, South Tehran Branch, Iran. A_salajegheh@azad.ac.ir

ABSTRACT
Service Based Applications (SBA) running in incongruous environments are subjects to varying constraints that can lead to fluctuations in the quality of the application, alternatively it is desirable to count on autonomic mechanisms to guide the adaptation of SBA according to changes in the computing infrastructure. Note that it is possible to have a set of different adaptation strategies applicable in the same situation. Context-aware mobile computing model is designed to automatically adapt its behavior to current contextual requirements and choose the best adaptation strategy according to the system requirements. This model provides all functionalities of MAPE-K model and uses Fuzzy theory in both analysis and plan phases to characterize adaptation necessity and choose best adaptation strategy in accordance with fundamental criteria.

Keywords: Context-awareness, Dynamic Adaptation, Fuzzy logic, Quality of service

INTRODUCTION
Mobile computing introduces new challenges in designing computer hardware and software due to user mobility, the varied types of devices used, resource constraints, and the dynamic nature in execution context. Therefore, adaptation and evolution of Service Based Systems (SBS) are of great importance in order to guarantee the quality of service (QoS) defined in a Service Level Agreements (SLAs) that are negotiated between the providers and users. Depending on the events and conditions triggering adaptation, there exist a set of adaptation strategies. Adaptation strategies define the possible ways to achieve adaptation requirements [1]. However, selection of the most suitable adaptation strategies is a complex task due to multiple criteria involved in the process of decision making [2].Strategy decision mechanisms [1], define the way a particular adaptation strategy is chosen based on the adaptation needs, SBS state, history of previous adaptation, etc. In this paper, we propose a fuzzy-based adaptation model that can be used in context-aware middleware. First of all we formulate the service adaptation process by using...
fuzzy linguistic variables and membership degrees to define the context situation and determine the adaptation necessity. Subsequently, we take advantage of the fuzzy logic too, for considering multiple criteria involved in the process of adaptation like security and cost to select the most suitable adaptation strategies. According to [1], adaptation strategies are categorized as: reconfigure, substitute or reselection, compense, re-plan, re-compose, and re-negotiate. In this model, we considered for common strategies involve: re-negotiate, re-selection, re-configure, and re-compose. The rest of this paper is organized as follows. First section gives an overview of the adaptation approaches. Next sections present the design of our model and then the process of adaptation is detailed step by step. Finally, conclusion and our future work.

**Approach Overview**

Adaptation of a service-based application refers to the capability of a system to continuously change itself in order to satisfy new requirements. Changes during operation may stem from software system’s self, such as failures or context, therefore, detecting significant changes, deciding how to react, and finally acting to execute decisions, are the essential steps to improve lifetime of systems. To tackle this problem, our model described by IBM’s MAPE-K reference model of an autonomic system [3]. When a change is detected, the monitoring phase triggers the next phase to analyze it and finds an adaptation strategy in case it is required. Then this strategy is given to the planning phase to compute a schedule of actions that will satisfy the strategy. The last step is the execution of the schedule to change the system (application, services and the environment). First of all, monitoring functions are used to provide an instructive and dynamic view of the adaptive entity and its environment. The model achieves context-awareness by using an event notification model, which monitors the status of the interested notification context and reports the event to the subscribed entities in the analysis phase. This model is taken from MobiPADS [4]. After this, FLC-QM (Fuzzy Logic Controller-Quality Manager) calculates the amount of QoS fluctuation due to the domain specific changes. Subsequently sends a message to Strategy Manager to choose the suitable adaptation strategy. After producing adaptation strategy by the analysis phase, the planner investigates available algorithms for the chosen strategy and if there is not any possible condition, it reports the problem to the analysis phase for producing other adaptation strategy. Otherwise, it will compute a schedule of actions that will satisfy the strategy. We give some details about these components and their functions in section 3.

**Context-aware Adaptive Model**

Figure 1 illustrates the macro-components of our proposed model, showing core components. We detail each step in the remaining parts of this section.

![Figure 1. Overview of proposed adaptive model](image-url)
model, which monitors the status of the interested context taken from MobiPADS [4]. The event notification model allows the composition of several primitive events to form a hierarchical composition of events in the form of an event graph. To facilitate the provision of this option, the event listener is required to supply a description of the conditions under which the listener will be informed of any change. In this connection, an event-condition object is created that keeps track of the status of the event source, and issues a notification only when the specified condition is fulfilled. For example, network bandwidth as a computational context, may change persistently. When the fluctuation of network bandwidth is inconsiderable, event-condition object debars informing analysis phase and accordingly, computational load of the process will not be increased unnecessarily. Context manager checks the sensor network as well as user profile regularly, and then abstracts the context information into event listener practicable form.

In the SOA domain, non-functional requirements are usually the result of a negotiation process engaged between the service provider and user, which culminate in the definition of a Service Level Agreement (SLA) concerning their respective obligations and expectations [5]. SLA-manager is responsible for managing the arrival/departure of a user with the associated SLA and observed variation in the SLA parameters of the constituent operations. Upon receiving a notification of a significant variation of the model parameters from monitoring phase, the model finds out whether an adaptation action must be performed. To this end, it calculates the effects of technical metrics changes on the QoS parameters by using a fuzzy inference system which is called FLC_QM. FLC_QM consists of two fuzzy inference engines, namely Performance assessment engine and User-satisfaction assessment engine.

The definition of quality parameters such as dependability, security and reputation has a high degree of ambiguity that makes it difficult to define them in a precise approach using crisp values.

Fuzzy logic is particularly suitable among other techniques in dealing with uncertainty and using imprecise parameters [6]. In this paper, we take advantage of the fuzzy logic for measuring overall QoS which is called User-Satisfaction that is the combination of availability and performance parameters. The definitions of the metrics for each quality attributes are mentioned in [7]. Figure 2 provides information about FLC_QM construction.

After realization of technical changes like network bandwidth variation by monitoring components, FLC-QM calculates the amount of response time and throughput according to the current value of connection network bandwidth. These two metrics are used as inputs for the performance assessment engine of FLC-QM.

Therefore, Performance assessment engine takes these metrics as inputs and applies the QoS rules from the knowledge database to provide the overall degree of the performance. As the next step, we set the values of performance and availability as inputs, with content User-satisfaction level as output for second inference engine. Defining the membership function for inputs and outputs are predetermined based on experts as discussed elsewhere [6, 8, 9]. Figure 3, illustrates fuzzy surface.
After determining the drop amount of user-satisfaction parameter, the FLC-QM sends a request to Decision Maker for selecting appropriate adaptation strategy. Depending on the adaptation triggers and domains of application usage, there exist a set of adaptation strategies. We consider two fundamental criteria for selecting the reasonable strategy: cost and security. To get a realistic and complete view of the decision process of service providers, the costs of adaptation need to be taken into account. Security is a broad concept that includes many aspects such as Confidentiality, Authenticity, Availability and Privacy. These are the main properties that are likely to be specified in user’s security policy [10]. To judge if an adaptation strategy like composition is more secure than another, we propose the concept of security impact factors to standardize security properties such as those mentioned based on some works like [10]. The security impact factors should be predetermined based on expertise. Note that in different scenarios the weight of these four properties may change.

There are different methods to calculate and optimize the cost and security of each adaptation strategy like [10, 11]. Figure 4, shows the construction of strategy manager’s fuzzy logic controller.

As we explained in the previous section, the analysis phase produces a strategy. According to [12], a strategy specifies a new state to reach for the service, the application or the environment (i.e. the system). The strategy doesn’t specify how to make the adaptation but rather what the goal of the adaptation action is. More precisely, it is the role of the planning phase to find a set of actions allowing the reconfiguration of the system. Until now the planning phase has acquired little consideration in the relation of adaptation and in many cases the planning algorithms used to produce elementary ordering of actions. Some works have been done in research topics such as artificial intelligence or control theory to build planning algorithms that allow computing schedules of actions [12]. Some of these algorithms are presented in [13, 14, 15, 16, 17]. Possibility of considering multiple algorithms for each adaptation strategy is an important advantage which raises the adaptation model flexibility and adaptability. Adaptation Strategy Generator is responsible for determining each strategy’s algorithms and selecting the best one. In the end, once the planning engine has computed the action plan from the strategy, the execution phase macro component is called to adapt the service, the application or the environment. Unlike the other phases, the execution phase cannot be generic, since it is in direct relation with the application.
implementation [12]. The Adaptation Manager is in charge of carrying out the adaptation actions at runtime. The adaptation manager carries on the decided modification to the service, or to a set of services, as indicated by planning components. One of the challenges in the execution phase is to ensure that the reconfiguration or adaptation actions will not make the application enter an unsafe state. The problem is left to the execution implementation [18].

According to [19], the division between planning and the knowledge used to effect adaptation in the autonomic MAPE-K loop is quite fuzzy, as one would expect. The knowledge in an autonomic system can come from sources as diverse as the human expert (in static policy based systems to logs that accumulated data from probes charting the day-to-day operation of a system to observe its behavior, which is used to train predictive models). In our model, Knowledge macro-component is accessed through the Data Access Library, which allows accessing the parameters describing the composite service and its operating environment (it includes the set of tasks in the abstract composition, the corresponding candidate operation with their QoS attributes, and the current solution of the adaptation problem that derives the composite service implementation).

CONCLUSION AND FUTURE WORKS
In this paper we have presented a novel approach for adapting SBAs against technical changes based on fuzzy logic in two phases of MAPE-K reference model. First, the degree of QoS can be gathered by changes of QoS parameters triggered by technical changes like network-bandwidth fluctuations. Second, decision making for adaptation can be inferred by trading off between adaptation strategies, their importance, preconditions, security and costs. Therefore, the model is able to choose the best between renegotiation, reselection, reconfiguration, and re-composition according to the requirements. We particularly used hierarchy fuzzy systems for the problem of rule explosion and to increase the scalability and reusability of applied rules. A first direction consists in dealing with reducing adaptation work load in repetitive cases. In this respect, using learning algorithms in analysis phase can help improving model performance and cost. Moreover, only technical triggers and QoS parameters are investigated in this paper. However, our approach is flexible in that its application is independent and can support adaptation with respect to changes of parameters from various categories.

REFERENCES
6. B. Pernici and S. H. Siadat, “Selection of Service Adaptation Strategies Based on
SOA Adaptation using Fuzzy Theory in Context-Aware Mobile Computing Model