2013 IEEE 6th International Conference on Service-Oriented Computing and Applications

A Model for Intelligent Adaptation and Evolution of Polymorphic Services

Anthony Karageorgos, Nikolay Mehandjiev
MBS
University of Manchester
Manchester, UK
{anthony.karageorgos,n.mehandjiev}@mbs.ac.uk

Elli Rapti
Institute of Research and Technology - Thessaly
CERTH
Volos, Greece
erapti@ireteth.certh.gr

***Abstract*—The highly dynamic nature and ubiquity of contemporary computer environments requires services to adapt and evolve to match varying contexts. Such services can be referred to as *polymorphic* since they can deliver their functionality in different forms. This paper proposes a fuzzy-based model for intelligent adaptation and evolution of polymorphic services based on context. Context is represented by parameters whose values fluctuate dynamically and their characteristics, such as range and mean, can evolve in time. Service adaptation is realized by selecting suitable service provision policies depending on context parameter configurations. The suitability of each service provision policy is determined by qualitative criteria which are estimated by fuzzy rules applied on context parameter values. Evolution is realized by having the fuzzy rule structure and parameters altered dynamically to align with evolved context. The applicability of the proposed approach is demonstrated in a traffic management case study.**

Keywords*—*Polymorphic Services; Service Adaptation; Service Evolution; Intelligent Service Maintenance;

I. INTRODUCTION

The advent of cloud computing and the resulting high interconnection and openness of software environments have increased the pace of user requirements change as well as the alterations of service provision contexts. Therefore the need for services to adapt dynamically to rapidly changing circumstances has become urgent [1] and as a result various approaches to service adaptation have been proposed [2, 3].

Additional issues in service provision arise from the fact that contexts can no longer be considered static and therefore adaptation methods cannot simply select services based on pre-recorded service-to-context mappings [3]. Contexts can often change and hence service adaptation needs to consider evolving contexts, a process which is commonly termed service evolution [4].

Service evolution refers to altering service functionality to match contexts and requirements not previously specified in exact form [5]. When services include both adaptation and evolution capabilities they can deliver their functionality in different forms. Such services can be termed *polymorphic services*, and it is envisioned that they will become increasingly important in highly dynamic environments such as internet of things and mobile clouds [6]. Current service-centric models largely lack the capability to continuously explore contextual changes that drive service evolution [4, 7].

To handle the vast amount of contextual information and numerous combinations of adaptive system services, fuzzy logic is often applied for service adaptation [8]. In fuzzy systems, evolution commonly refers to progressive modification of their fuzzy rules [9]. Therefore, it can be argued that fuzzy service evolution can be achieved through adaptive evolution of the respective fuzzy rule-base, for example as suggested in [10].

Based on this view, a model for fuzzy based adaptation and evolution of polymorphic services according to context is proposed in this paper. The model uses fuzzy variables and fuzzy membership functions to represent context situations and criteria for adopting service execution policies as proposed in [11]. Each service execution policy is characterized by a fitness level calculated based on the distance between optimum and current context situation. Service adaptation is then achieved by selecting the policy having the highest fitness. In contrast to other rule based service adaptation methods which alternate between rule sets [12], the associated fuzzy rule-bases evolve dynamically by recursive optimization of rule structure and parameters based on clustering [13].

In Section II of this paper the necessary background and motivation for polymorphic service adaptation and evolution is provided. The problem is formally defined and the solution approach is outlined in Section III. A fuzzy-based model for adaptation and evolution of polymorphic services is described in Section IV, followed by a demonstration of its applicability in a traffic management case study in Section V. Section VI discusses relevant work and Section VII concludes the paper.

II. MOTIVATION - BACKGROUND

The dynamic nature of business environments requires service-based applications to be highly reactive and adaptive since they are subject to constant changes and variations [14]. Therefore it is argued that services should be equipped with mechanisms to ensure that they are able to adapt to meet changing requirements [15], and evolve due to changes in structures, in behavior and policies consisting their contextual environment [7]. Contextual changes can be detected by monitoring service application execution and the respective user input and environmental parameters [16]. Such services

This research is supported by EU Marie-Curie Intra-European Fellowship for Career Development grant No. PIEF-GA-2011-301125.

978-1-4799-2701-2/13 $31.00 © 2013 IEEE
DOI 10.1109/SOCA.2013.47

capable of exhibiting both adaptation and evolution to cope with circumstances that were not exactly specified in advance are referred to in this paper as *polymorphic services*.

Service adaptation is often achieved by applying certain strategies during execution to modify service functionality. An adaptable service application should then continuously monitor and modify the applicable strategies on the basis of changes in its environment [17]. Furthermore, software evolution refers to gradual and constant change of service functionality in particular situations over a period of time [18]. Service evolution can therefore be seen as a gradual response to context-aware feedback received from the service environment.

Current context aware service-centric software models do not adequately support both adaptation and evolution of services, and very often tuning and adjustment are called adaptation, self-organization or even evolution [1]. Work in service adaptation has reached a level of maturity [19] and currently an issue of concern is the selection of appropriate adaptation mechanisms given the complexity and dynamism of the service environment [20]. On the other hand work in service evolution currently focuses on ensuring the compatibility between evolved services and service clients [5] and little attention is paid on modelling the ambiguity of service environments.

Fuzzy techniques are a promising approach for dealing with the impreciseness of service environments. The view adopted in current fuzzy-based service adaptation approaches considers that the fuzzy rule structure and parameters should be fixed [21]. However, since contextual service environment data are generally non-stationary and continuously change, it is reasonable to expect that the structure of the respective data models need be also dynamic and evolving. Thus, the structure and parameters of fuzzy rules used to model contextual data would generally be not fixed but gradually evolving as more data enter the system. This view is adopted in the approach followed in this paper which models the dynamism and uncertainty of the service context using evolving fuzzy rules.

III. Problem Specification and Solution Approach

A. Problem Statement

A brief explanation of relevant background concepts is provided below together with the respective formal definitions.

Definition 1 (Context State): A context state cs is a tuple comprising a combination of context parameter values. We denote it as: $cs = \langle c_i \rangle, c_i \in D_i, i = 1..n$, where D_i is the domain of the respective context parameter.

Definition 2 (Context): An overall context CS is the set of all possible context states that can be formed from the domain values of a number of context parameters. That is: $CS = \{\langle c_i \rangle, c_i \in D_i, i = 1..n\}$ where D_i is the domain of the respective context parameter. A context C is any subset of an overall context CS, that is $C = \{cs: cs \in CS\}$. C is termed a sub-context of CS and CS is considered a super-context of C.

The concept of context generally refers to a set of interrelated conditions in which something can exist or occur. In this respect context in service oriented systems refers to the external conditions that affect the provided service functionality, for example in low network bandwidth it would be mandatory to have an e-mail client not transmitting e-mail attachments but instead passing only the message header or body in text format. In the proposed model context conditions are represented with a number of context parameters. The values of context parameters can be directly measured from the system environment, for example via appropriate sensors.

A context state then is a snapshot of a service context and it is described by specific values of the respective context parameters, such as level of network bandwidth or location coordinates.

Definition 3 (Context Evolution): Context evolution is a mapping between sub-contexts of an overall context CS via an evolution function ev, that is $ev: 2^{CS} \rightarrow 2^{CS}$. The term evolved context, denoted by $ev(C)$, is used to refer to context mapped from a context C via a context evolution function ev.

Context can evolve over time, for example the capacity of wireless networks supporting mobile clouds. In such cases installation of new equipment can permanently increase bandwidth levels enabling a new range of software capabilities.

Definition 4 (Service Execution Policy): A service execution policy p is a method with which a service s can be executed to achieve certain functional and non-functional characteristics.

Policies can affect various characteristics such as resource usage or the quality of service provided. In the above e-mail client example the omission of attachments in message transmission in low bandwidth conditions would be ensured by selecting an appropriate execution policy.

Definition 5 (Policy Fitness): Fitness f of a service execution policy p for a context state cs with respect to executing service s is a number in [0,1] describing how suitable the policy p is for executing service s in context state cs. It can be defined as a mapping $f: C \times P \rightarrow [0,1]$ where C is a context and P is a set of execution policies suitable for service s.

Definition 6 (Adaptation Function): Adaptation function of a service s, denoted by $adpt$, is a mapping $adpt: C \longrightarrow P$, where C is a context and P is a set of execution policies suitable for executing service s. The family of adaptation functions of s for all subsets of an overall context CS is denoted by $ADPT_{CS}$, that is: $ADPT_{CS} = \{adpt: C_j \rightarrow P, C_j \subseteq CS, j = 1..|2^{CS}|\}$.

Definition 7 (Selection Function): Selection function Sel of a service s is a mapping between sub-contexts of an overall context CS and the respective family of adaptation functions $ADPT_{CS}$ of s, that is $Sel: 2^{CS} \rightarrow ADPT_{CS}$.

Definition 8 (Adaptive Service): A service s is adaptive in context C, denoted by $adapts(s, C)$, when an adaptation function $adpt$ can be selected via a selection function Sel such that for every context state cs belonging to C a policy p_{opt} can be determined via $adpt$ such that its fitness f is maximum for cs, that is:

$$adapts(s, C) \Leftrightarrow \exists Sel: 2^{CS} \longrightarrow ADPT_{CS}, adpt = Sel(C),$$

$$\forall cs \in C, p_{opt} = adpt(cs), f(cs, p_{opt}) = \max_{p \in P} f(cs, p)$$

where P is a set of execution policies suitable for executing service s and $ADPT_{CS}$ is the family of adaptation functions of s for CS sub-contexts.

Definition 9 (Evolving Service): A service s is evolving in context C, denoted by $evolves(s, C)$, when s is adaptive in every context $ev(C)$ that can evolve from C. In other words a service s is evolving in context C if it is adaptive in every sub-context of the super-context CS of C. That is:

$$evolves(s, C) \Leftrightarrow \forall ev(C) \in CS, adapts\big(s, ev(C)\big)$$

or in other words:

$$\begin{aligned} evolves(s, C) \Leftrightarrow \exists Sel: 2^{CS} \longrightarrow ADPT_{CS}, adpt \\ = Sel(ev(C)), \forall cs \in ev(C), p_{opt} \\ = adpt(cs), f(cs, p_{opt}) = \max_{p \in P} f(cs, p) \end{aligned}$$

where CS is the super-context of C and $ADPT_{CS}$ is the respective family of adaptation functions of s.

Definition 10 (Polymorphic Service): A service s is polymorphic in context C, denoted by the predicate $isPolymorphic(s, C)$, iff it is evolving in C. That is:

$$isPolymorphic(s, C) \Leftrightarrow evolves(s, C)$$

From the above definition it can be easily derived that when a service s is polymorphic in a context C it is also polymorphic in all sub-contexts of the super-context CS of C including CS itself. That is:

$$\begin{aligned} isPolymorphic(s, C) \Leftrightarrow \forall C \subseteq CS, isPolymorphic(s, C) \\ \Leftrightarrow isPolymorphic(s, CS) \end{aligned}$$

Hence to develop polymorphic behaviour for a service s in evolving sub-contexts C of an overall context CS appropriate selection and adaptation functions need to be provided to ensure the fittest execution policy will be applied for s in any possible context state cs of CS. In the proposed model this problem is addressed by considering a generic family of adaptation functions. These functions calculate policy fitness based on qualitative context parameters, which are estimated using fuzzy rules applied on context state values. Context-based adaptation function selection is achieved by having the structure and parameters of fuzzy rules modified dynamically following evolved context, resulting thus in different adaptation functions for different sub-contexts.

B. Polymorphic Service Adaptation and Evolution

Selection of the most suitable service adaptation strategies is a complex task due to the multiple criteria involved [22]. For example, different adaptation actions have different time and computation complexity. Furthermore, adaptation criteria obtained from an application context are often erroneous and vague [8] making therefore exact adaptation methods unsuitable for contexts involving several parameters as is the case in open system settings. Finally, context dynamically evolves and therefore it is argued that context evolution should be continuously taken into account by considering "context as process" instead of only "context as state" [23].

To tackle context complexity and uncertainty several authors suggest the use of fuzzy logic to model context parameters [17, 24]. Along this line we draw inspiration from [11] and we model context characteristics using fuzzy quality variables and subsequently use them to calculate the fittest service execution policy. To capture context evolution the fuzzy quality membership functions are not described by static formulae but instead they are obtained from the consequent parts of fuzzy rules which are calculated dynamically based on incoming context parameter values. These fuzzy rules are evolving by having the centres of their antecedent part fuzzy membership functions changing dynamically according to context parameter value fluctuations.

The steps of the proposed polymorphic service adaptation and evolution process are depicted in Fig. 1. The process includes three main stages, that is: a) Fuzzification of context parameters and estimation of fuzzy context quality values, b) Adaptation to context which involves calculation of the fitness degree of each relevant execution policy and subsequent fittest policy selection, and c) Alignment with evolving context, which involves evolution of quality membership functions.

IV. Fuzzy-based Adaptation and Evolution Model

A. Estimation of Context Quality

A main premise of the proposed model is that adaptation to context should be driven by qualitative contextual characteristics determined by the values of measurable context parameters. For example, if a videoconference application executing over a mobile network is deciding whether to enable full multimedia support for the participants, this will be commonly dependent on the overall network quality, which is determined by a number of network parameters such as network bandwidth and network latency. Hence the concept of context quality is introduced.

Definition 11 (Context Quality): A context quality q is a parameter characterizing a service context which cannot be measured directly from the service environment.

Context qualities are generally qualitative variables and they are dependent on context parameters which are directly measurable from the service environment. To better represent their qualitative aspects context qualities are considered as fuzzy variables that take fuzzy values represented by linguistic terms. For example, the context quality $customerType$ can take the fuzzy values 'good', 'average' and 'bad'. Furthermore, to calculate the membership function value of each linguistic term of a context quality q we utilise fuzzy rules of the form:

$$\text{IF } cp_1 \text{ is } LV_1 \text{ AND } \ldots \text{ AND } cp_k \text{ is } LV_k \text{ THEN q is } QV \quad (1)$$

where $LV_1, \ldots LV_k$ are linguistic terms corresponding to context parameters $cp_1, \ldots cp_k$ and QV is a linguistic term of the context quality q. For example, to obtain the degree of a customer type being 'good' based on the customer level of orders and credit we could have the following fuzzy rule:

$$\text{IF } orderLevel \text{ is } HIGH \text{ AND } credit \text{ is } LOW \text{ THEN } customerType \text{ is } GOOD$$

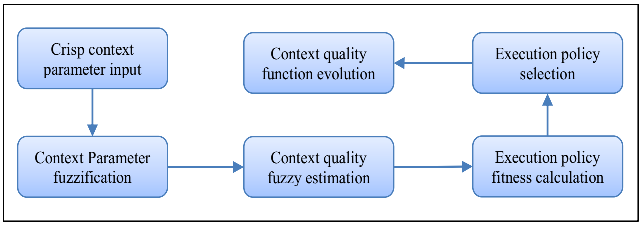

Fig. 1. Polymorphic Service Adaptation and Evolution Process iterates at regular time intervals or when context parameter value changes are detected.

In the general case more than one fuzzy rules can correspond to each linguistic term of a context quality and they can be aggregated to obtain the fuzzy result using a common fuzzy rule aggregation method.

Context quality can be quite variable due to the dynamism of the service environments. For example, in the videoconference scenario mentioned above the network quality can vary according to network availability at different user locations. Furthermore, context variation affects the suitability of service execution policies at any given time. For this reason it is argued that the time intervals in which the values of context qualities can be considered as stable should be identified, and for each such time interval these stable values should be used for service execution policy determination. The start of such time intervals can be indicated by context state changes. For example in a context described by arriving customer characteristics the context state remains stable for the whole duration that a service is provided to a particular customer until his departure. When the context state changes are continuous, as is the case in the videoconferencing example for instance, it is more practical to calculate context quality and subsequently apply adaptation methods at time points indicating regular time intervals.

Based on the above arguments the concept of context quality snapshot is introduced.

Definition 12 (Context quality snapshot): A context quality snapshot qs of a context C is a set of triples each comprising a fuzzy variable representing a context quality, a corresponding linguistic term, and the value of the respective membership function at a particular time point. That is:

$$qs_t = \{\langle q_i, QV_{q_i}^{v_i}, \mu_{QV_{q_i}^{v_i}}(t)\rangle, q_i \in Q_C, QV_{q_i}^{v_i} \in QV,$$
$$v_i = 1..lv_i, i = 1..|Q_C|\}$$

where q_i is the i^{th} context quality of context C, $QV_{q_i}^{v_i}$ is the v_i^{th} linguistic term of context quality q_i, Q_C is the set of all context qualities characterising context C, QV is a set of fuzzy linguistic terms, lv_i is the number of linguistic terms corresponding to the i^{th} context quality, $|Q_C|$ is the number of context qualities in Q_C, and $\mu_{QV_{q_i}^{v_i}}(t)$ is the value of the respective fuzzy membership function "q_i is $QV_{q_i}^{v_i}$" at time point t, indicating the degree at which q_i can be considered as equal to $QV_{q_i}^{v_i}$ at time point t. We refer to $\mu_{QV_{q_i}^{v_i}}(t)$ of each triple as "active membership value of linguistic term $QV_{q_i}^{v_i}$ with respect to executing s with policy p in context C".

B. Policy Fitness

Fitness calculation is based on the concept of fuzzy distance of a policy p and the context quality snapshot qs_t at a selected time point t. In this paper the concept of fuzzy distance of a policy p is defined in a similar manner as in [11] considering the sum of differences between current and "best" membership function values for all fuzzy terms of all context qualities that affect p fitness. For example if there is only the context quality $customerType$ that can take only the fuzzy term 'good' and we have $\mu_{GOOD}^{BEST}(p)$=0.8 and $\mu_{GOOD}^{NOW}(p)$=0.6 then the fuzzy distance between p and the current qs_t is 0.2.

The fitness of a policy p with respect to executing a service s at time point t is then obtained from the following formula:

$$f_p(t) = \frac{1/d_p(t)}{\sum_{k=1}^{|P|}\left(1/d_{p_k}(t)\right)} \tag{2}$$

where $d_p(t)$ is the fuzzy distance of policy p and context quality snapshot qs_t at timepoint t, P is the set of policies that can be apply to execute s, $|P|$ is the number of policies in P and $\sum_{k=1}^{|P|}\left(1/d_{p_k}(t)\right)$ is the sum of fuzzy distances of all policies in P.

C. Support for Context Evolution

Contexts can evolve, for example a context C_1 can be fully replaced by another context C_2, both of the same super-context CS. In the general case context parameter values have different ranges at different contexts. For example, customer orders and credit can be on average higher or lower depending on the financial status of the country of customer origin. As another example, network bandwidth availability can be reduced due to an increase in user base or due to some equipment malfunction.

In some cases such changes in the ranges of context parameter values do not require any changes in service provision and hence any adaptation criteria already in place remain valid. For example, if network bandwidth is generally decreased then it will be correct to completely disable full multimedia videoconferencing, as would have been dictated by existing service adaptation criteria. Therefore, in such cases context evolution does not require any respective service evolution.

In other cases, however, addressing context evolution requires evolution of service provision. For example, in a financial crisis all customer transaction amounts fall, yet those customers with the highest transactions could still be considered as 'good' customers and be entitled to receive benefits, such as discount. Considering the example given in Section IV.A applying this strategy would require changing the way the membership function of the fuzzy value 'good' of context quality $CustomerType$ is calculated to incorporate the decrease in 'good' customer transaction levels.

To capture the evolving element of context quality fuzzy values it is considered that the membership functions of the antecedent part of fuzzy rules used to estimate context qualities evolve as well. To model this evolution first the fuzzy sets of the antecedent part linguistic terms are represented using fuzzy numbers. Subsequently, a method for dynamic modification of the antecedent part rule centres according to evolving context changes is provided. This method is described in Section IV.D.

For each context quality a fuzzy rule base including all fuzzy rules used to estimate the respective context quality fuzzy values is considered. For the estimation of each such value more than one fuzzy rules are generally employed. Each fuzzy rule base comprises rules of the form:

$$\text{IF } cp_1 \text{ is } cp^*_{ik1} \text{ AND } \dots\ cp_m \text{ is } cp^*_{ikm} \text{ THEN } q \text{ is } QV_k \quad (3)$$

where $cp_j, j = 1..m$ are the input context parameter values, $cp^*_{ikj}, j = 1..m$ are the centres of the respective fuzzy sets representing fuzzy numbers, QV_k are linguistic terms of context quality q, $k = 1..L_q$, where L_q is the number of linguistic terms of context quality q, and $i = 1..R_{qk}$, where R_{qk} is the number of fuzzy rules used to calculate the linguistic term QV_k of context quality q. For brevity input context parameters and fuzzy rule centres are represented in vector notation, that is $\overline{\mathbf{cp}}$ and $\overline{\mathbf{cp}}^*_{ik}$ respectively. Such vectors can be considered as representing points in an *m*-dimensional space and therefore in the following the terms input context parameter vector and input data point are used interchangeably.

Furthermore, based on the approach introduced in [13] for the antecedent rule parts of the rules described in (3) we consider Cauchy-type membership functions of the form:

$$\mu_{ikj}(cp_j) = e^{-a\left\|cp_j - cp^*_{ikj}\right\|^2} \quad (4)$$

where cp_j is the j^{th} input parameter, cp^*_{ikj} is the center of the respective j^{th} fuzzy set, $a = 4/r_a^2$, and r_a is a positive constant.

Based on (3) and (4) and assuming algebraic product as the fuzzy ADD operator the firing level of each rule i specifying the membership function $\mu_{ik}(q)$ of linguistic term QV_k of context quality q is defined as follows:

$$\mu_{ik}(q) = \bigcap_{j=1}^{m} \mu_{ikj}(cp_j) = e^{\sum_{j=1}^{m} -a\left\|cp_j - cp^*_{ikj}\right\|^2} = e^{-a\left\|\overline{\mathbf{cp}} - \overline{\mathbf{cp}}^*_{ik}\right\|^2} \quad (5)$$

where $\mu_{ikj}(cp_j)$ is the membership function of the respective fuzzy set, $\overline{\mathbf{cp}}$ is the vector of input context parameters, $\overline{\mathbf{cp}}^*_{ik}$ is the vector of antecedent part fuzzy sets and $a = 4/r_a^2$, where r_a is a positive constant acting as normalization factor.

There are many ways to aggregate rule outcomes but for simplicity it is considered that for each linguistic term QV_k of a context quality q the respective membership function value $\mu_k(q)$ is calculated as a simple average of the corresponding values $\mu_{ik}(q)$ of all fuzzy rules that imply QV_k. That is:

$$\mu_k(q) = \frac{\sum_{i=1}^{R_{qk}} \mu_{ik}(q)}{R_{qk}} \quad (6)$$

where $\mu_{ik}(q)$ is the membership function of the linguistic term QV_k corresponding to the i^{th} rule, and $i = 1..R_{qk}$ where R_{qk} is the number of fuzzy rules used to calculate the linguistic term QV_k of context quality q.

Support for context evolution then consists of having the antecedent fuzzy rule centers cp^*_{ikj} changing dynamically in time due to evolution of context parameter values. This results in evolution of the ranges of the membership function values $\mu_k(q)$ for all linguistic terms QV_k of a context quality q. This in turn affects the calculation of policy fitness since at each time point t the calculated membership function value $\mu_k(q)$ is the active membership value $\mu_{QV^k_q}(t)$, which is used in (1) to calculate the fuzzy distance of policies to current quality snapshot, and thus influences the policy fitness values obtained by (2).

D. Evolving Fuzzy Rules

A common approach to fuzzy rule evolution is that of clustering incoming data points dynamically [25] and considering cluster points as antecedent part rule centres. Subsequently, variations in incoming data values result in identifying new cluster points which in turn replace previous antecedent part rule centers [9] and hence result in rule evolution. Modified fuzzy rules result in different ranges of calculated fuzzy values of the consequent rule parts.

In this paper a method for fuzzy rule evolution which implements dynamic rule center modification based on the concept of data point potential [13] is proposed. As mentioned in Section IV.C the values of incoming context parameters appearing in the antecedent part of fuzzy rules are considered as coordinates representing data points. Assuming a set CP of data points then the potential P of a data point $\overline{\mathbf{cp}} \in CP$ is a measure describing the suitability of $\overline{\mathbf{cp}}$ of being a cluster center and it depends on the Euclidean distances between $\overline{\mathbf{cp}}$ and all other data points [13]. That is [26]:

$$P = \frac{1}{N}\sum_{j=1}^{N} e^{-a\left\|\overline{\mathbf{cp}} - \overline{\mathbf{cp}}_j\right\|^2} \quad (7)$$

where P is the potential of the data point $\overline{\mathbf{cp}}$ to be a cluster centre, N is the number of data points considered, and $a = 4/r_a^2$ where r_a is the required cluster radius. From the above formula it becomes clear that the closer a data point is to most other points the higher is its potential of being a cluster center. Based on successive calculations of data point potential, when new data points enter the system dynamically, the centers of antecedent rule parts can be adjusted resulting thus in fuzzy rule evolution [9, 10].

A similar method for adjusting the centers of antecedent parts of the fuzzy rules used to estimate context qualities is proposed. However, contrast to general dynamic clustering approaches, where the number of rules is variable and depends on clustering process results [21], a fixed number of rules that have initially been drawn by human experts is considered.

The proposed approach maintains a rule base for each context quality q comprising R_{qk} rules for each linguistic term QV_k, or $\sum_k(R_{qk})$ rules in total. All rules are assumed to take input values belonging to the same context parameters. Each time a new data point $\overline{\mathbf{cp}}$, that is a tuple of context parameter values that are used as input in the respective fuzzy rules of context quality q linguistic terms, is recorded the current rule centres $\overline{\mathbf{cp}}_{\mathbf{i}}^{*}, i = 1..R_{qk}$ are assessed in comparison with the new data point and if appropriate they are replaced.

Assessment is done based on the potential of incoming data points. We denote with P_i^* and P the potential of rule centers $\overline{\mathbf{cp}}_{\mathbf{i}}^{*}$ and the new data point $\overline{\mathrm{cp}}$ respectively. A common criterion is when $P > P_i^*$ and $|P - P_i^*| = \max_j |P - P_j^*|$, with $P > P_j^*$, then to replace $\overline{\mathbf{cp}}_{\mathbf{i}}^{*}$ with $\overline{\mathbf{cp}}$. This means that a new data point will replace the cluster center from which it has higher potential and the highest potential difference from all other rule centres. The reason for commonly applying this fuzzy rule evolution criterion is to follow the rule base structure initially specified by the experts. Alternative rule replacement criteria can be considered as well. The process is depicted in Fig. 2.

The potential of each data point depends on the set of data points against which its distance is calculated. In a dynamic service system new data points appear at short time intervals and hence considering all previous data points in potential calculation becomes inefficient. For this reason the technique of moving window introduced in [26] is applied considering only the data points that have appeared in the most recent time window of a specified length (Fig. 3).

Let w be the length of the time window which considering that a data point is observed at each time step can be viewed as a list including w data points. Let P_w^w be the potential of data point $\overline{\mathbf{cp}}_{\mathbf{w}}$ at time w. Hence from (7) we have:

$$P_w^w = \frac{1}{w}\sum_{j=1}^{w} e^{-a\left\|\overline{\mathbf{cp}}_{\mathbf{w}}-\overline{\mathbf{cp}}_j\right\|^2} \tag{8}$$

where $a = 4/r_a^2$, where r_a is the required cluster radius. Those data points $\overline{\mathbf{cp}}_j$ located at distance further than r_a from $\overline{\mathbf{cp}}_{\mathbf{w}}$ do not contribute significantly in the P_w^w final value.

At time $w + 1$ the $\overline{\mathbf{cp}}_{\mathbf{1}}$ will be removed from the time window and $\overline{\mathbf{cp}}_{w+1}$ will be added. Hence the new potential P_w^{w+1} will be:

$$P_w^{w+1} = P_w^w + \frac{1}{w}\left(e^{-a\left\|\overline{\mathbf{cp}}_{\mathbf{w}}-\overline{\mathbf{cp}}_{w+1}\right\|^2} - e^{-a\left\|\overline{\mathbf{cp}}_{\mathbf{w}}-\overline{\mathbf{cp}}_{\mathbf{1}}\right\|^2}\right) \tag{9}$$

Equation (9) can be used to simplify potential recalculation when new data points appear in the system, that is when context state changes occur. It should be noted here that the removed point $\overline{\mathbf{cp}}_{\mathbf{1}}$ is not always the earliest data point in the time window.

For example the removal criterion may require that $\overline{\mathbf{cp}}_{\mathbf{1}}$ is not centre in any fuzzy rule of those used to estimate context qualities. Finally, the potential P_{w+1}^{w+1} of the new data point $\overline{\mathbf{cp}}_{w+1}$ is calculated from (8) as normal.

V. Polymorphic Travel Management Services

The proposed approach was applied in the problem of transport congestion management in urban areas, which represents an omnipresent yet increasingly serious problem

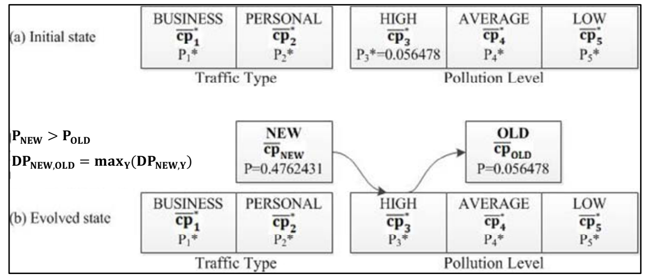

Fig. 2. Evolution of context quality rule bases for context qualities *Traffic_Type* and *Pollution_Level*. New rule center P_{NEW} has higher potential than the replaced point P_{OLD} and the maximum potential difference with all centers $\overline{\mathrm{cp}}_{Y}$ for which holds $P_{NEW} > P_Y$.

since transport congestion problems contribute about 70% of pollutants to urban environments. The MODUM EU project [27] develops a solution to this problem which involves balancing the routes drivers select by communicating with them on their intelligent mobile devices.

The underlying principle is that drivers are guided by personal assistant agents, which choose the “cheapest” route to the destination by contacting a special “pricing” service for each potential route. Drivers have an annual allowance of “money” to spend on travelling, and money left over at the end of the year are converted in some sort of benefit encouraging sustainable behavior. The pricing service considers numerous criteria including the pollution level currently in the atmosphere, the level of congestion, for example due to road works, and the number of vehicles currently using this route. Charging is performed according to scales, each corresponding to different approximate settings of the above criteria. For example, when there is high level of congestion at a particular route then a high charge scale is applied to prevent more drivers from using that route. However, when there is a longer term increase of a parameter, such as increase of congestion levels due to road works, then the strategy followed is to reduce the weight of the increased parameter values when selecting a charge scale since drivers have fewer route alternatives to choose from.

The above situation was modelled considering two context qualities. The first one is $Pollution$ which depends on the context parameters *Congestion level*, measured at expected delay in min/km, and *PSI index* which describes the atmospheric pollution and takes positive values, with values above 300 being hazardous. The second one is *TrafficType* which depends on the context parameters *CarNum* and *TruckNum* representing the percentages of private and professional vehicles respectively out of the total vehicles in that route at the time.

Three service policies controlling the pricing service were considered, corresponding to three pricing scales, *ScaleA* (low scale), *ScaleB* (normal scale), *ScaleC* (high scale). The proposed model was applied to select the best (fittest) policy for the execution of the pricing service at regular time intervals. The fuzzy rules used to calculate policy fitness evolved under the changing context. In the example considered in this paper, rule evolution was caused by significant progressive changes in the average level of pollution, so that to

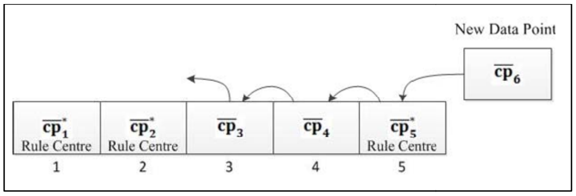

Fig. 3. Time window update process. The earliest non rule center data point is removed and all subsequent data points are advanced. Finally, the new data point is added at the last position.

progressively change the level of pollution that would be considered as normal. The example data set is depicted in Fig.4(a) and demonstrates a substantial yet reversible increase of pollution (*PSI* level), due to industrial accident for instance.

The results of applying the proposed model to the example data are presented in Fig. 4(c), and contrasted with the results depicted in Fig. 4(b), where neither adaptation nor evolution was used. Fig. 4(b) shows that when the adaptation procedure does not incorporate evolution, the fitness values show significant disturbances. Under the case of our approach, when evolution and adaptation were applied, Fig.4(c) shows that the policy fitness fluctuation is based on the overall input data differences as before, and it is not significantly affected by the increasing or decreasing trend in *PSI* context parameter values.

VI. Relevant Work

A. Service Adaptation

Adaptation has commonly been a fundamental aspect in service-oriented architectures. [28] proposes an architecture supporting dynamic service adaptation. That approach has limitations with respect to preserving the state of replaced service components. In [29] an adaptation machine that uses a set of mapping rules to manipulate messages exchanged between services is introduced. However, it cannot be determined whether a given set of mapping rules is sufficient to fully match descriptions of exchanged messages.

In [30] the focus is on defining flexible adaptation policies using contextual environments but adaptation is limited to two software components. In [31] an approach for unanticipated service adaptation is proposed based on the use of profile models. However, profile update still requires human operator intervention. [3] proposes an approach for service adaptation driven by service client queries, which generates services adapted to client functional requirements in a fully automated manner. However, it does not support queries including specifications of the required service behavior.

Fuzzy logic has often been employed to develop service adaptation models aiming to capture imprecise processes, for example [11] and [17]. Along this line Pernici and Siadat in [24] propose a fuzzy service adaptation approach which is based on describing QoS properties using fuzzy parameters aiming to achieve flexibility in service specification. Furthermore, in [32] a mixed approach to context inference combining clustering techniques and semantics is proposed, which improves rule engine reasoning at the expense of processing power requirements. However, none of the above approaches adequately support service evolution.

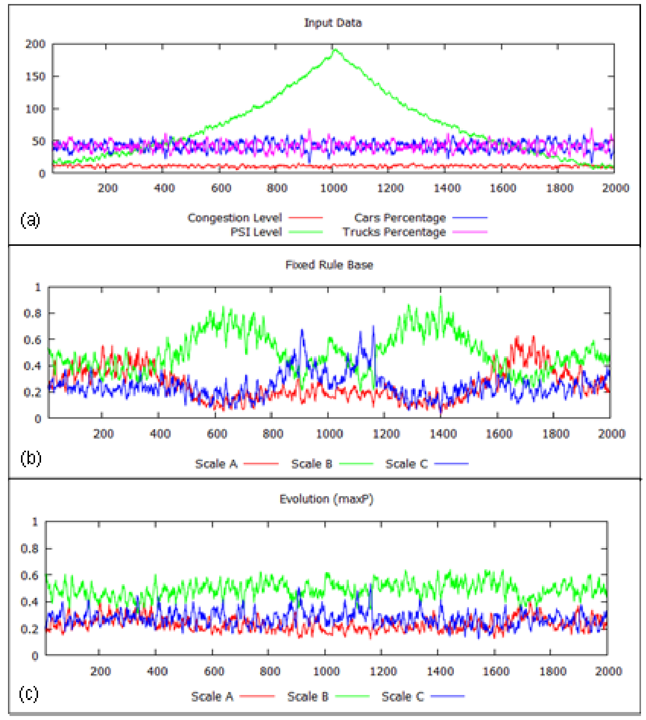

Fig. 4. Moving average values: (a) Input data, (b) Policy fitness when there is no evolution, and (c) policy fitness when there is evolution based on maximum potential difference. Input data trends result in significant policy fitness value differences when there is no evolution.

B. Service Evolution

Service evolution has been often realized by applying bio-inspired mechanisms such as in [33]. Furthermore, in [34] evolution is achieved by representing services by software agents that interact autonomously in a distributed environment. The majority of the research efforts has so far concentrated on the evolution of a single service. In this respect emphasis has been given on the taxonomy of evolutionary changes [7, 35], and on addressing the incompatibility using different tactics, including versioning [5], design pattern/adaptor [36-38] and model/theory/contract [39, 40]. However none of the above approaches sufficiently addresses the imprecision of the service environment since they all use exact models to represent service parameters.

VII. Conclusions

In this paper, a model for service adaptation and evolution of polymorphic services was introduced. The approach involves selecting suitable service execution policies according to context qualitative criteria which are estimated using fuzzy rules. Context evolution is captured by having the fuzzy rule parameters dynamically change following evolving context.

Further research is needed regarding how the evolution results are influenced by the size of the time window and by the weighting factor of fuzzy distance sum factors. In addition, further experimentation with large data sets is required to examine the performance of the method in different types of context parameter value ranges.

REFERENCES

[1] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl, "A journey to highly dynamic, self-adaptive service-based applications," *Automated Software Engineering,* vol. 15, pp. 313-341, 2008.

[2] C. Soo Ho, L. Hyun Jung, and K. Soo Dong, "A Comprehensive Approach to Service Adaptation," in *IEEE International Conference on Service-Oriented Computing and Applications (SOCA '07)*, 2007, pp. 191-198.

[3] A. Brogi and R. Popescu, "Service Adaptation Through Trace Inspection," *International Journal of Business Process Integration and Management,* vol. 2, p. 9, 2007.

[4] C. K. Chang, J. Hsin-yi, M. Hua, and K. Oyama, "Situ: A Situation-Theoretic Approach to Context-Aware Service Evolution,", *IEEE Transactions on Services Computing,* vol. 2, pp. 261-275, 2009.

[5] M. P. Papazoglou, S. Benbernou, and V. Andrikopoulos, "On the Evolution of Services," *IEEE Transactions on Software Engineering,* vol. 38, pp. 609-628, 2012.

[6] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems,* vol. 29, pp. 84-106, 2013.

[7] M. Papazoglou, "The Challenges of Service Evolution," *Advanced Information Systems Engineering,* p. 15, 2008.

[8] B. Pernici and S. H. Siadat, "Selection of Service Adaptation Strategies Based on Fuzzy Logic," *2011 IEEE World Congress on Services (SERVICES),* p. 8, 2011.

[9] A. Lemos, W. Caminhas, and F. Gomide, "Evolving Intelligent Systems: Methods, Algorithms and Applications," in *Emerging Paradigms in Machine Learning*. vol. 13, S. Ramanna, L. C. Jain, and R. J. Howlett, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 117-159.

[10] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," *IEEE Transactions on Fuzzy Systems,* vol. 10, pp. 667-677, 2002.

[11] J. Cao, N. Xing, A. T. S. Chan, Y. Feng, and B. Jin, "Service Adaptation Using Fuzzy Theory in Context-Aware Mobile Computing Middleware," in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*, 2005, p. 6.

[12] J. Yu, J. Han, Q. Sheng, and S. Gunarso, "PerCAS: An Approach to Enabling Dynamic and Personalized Adaptation for Context-Aware Services," in *Service-Oriented Computing*. vol. 7636, C. Liu, H. Ludwig, F. Toumani, and Q. Yu, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 173-190.

[13] S. L. Chiu, "Fuzzy Model Identification based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems,* vol. 2, p. 12, 1994.

[14] J. Hielscher, R. Kazhamiakin, A. Metzger, and M. Pistore, "A Framework for Proactive Self-Adaptation of Service-based Applications Based on Online Testing," *Towards a Service-Based Internet, Lecture Notes in Computer Science* vol. 5377, p. 12, 2008.

[15] F. Leymann, S. Dustdar, M. P. Papazoglou, and P. Traverso, "Service-Oriented Computing: A research roadmap," *International Journal of Cooperative Information Systems,* vol. 17, pp. 223-255, 2008.

[16] C. Zeginis and D. Plexoudakis, "Web Service Adaptation: State of the Art and Research Challenges," *Technical Report 410, ICS-FORTH,* p. 66, 2010.

[17] R. Cheung, G. Yao, J. Cao, and A. Chan, "A Fuzzy Service Adaptation Engine for Context-Aware Mobile Computing Middleware," *International Journal of Pervasive Computing and Communications,* vol. 4, p. 20, 2008.

[18] R. T. Mittermeir, "Software Evolution: Let's Sharpen the Terminology Before Sharpening (out-of-scope) Tools," *Proceedings of the 4th International Workshop on Principles of Software Evolution,* p. 8, 2001.

[19] S. Lane, Q. Gu, P. Lago, and I. Richardson, "Towards a framework for the development of adaptable service-based applications," *Service Oriented Computing and Applications,* pp. 1-19, 2013/07/14 2013.

[20] S. H. Siadat, A. Zengin, A. Marconi, and B. Pernici, "A fuzzy approach for ranking adaptation strategies in CLAM," in *Proceedings of the 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2012, pp. 1-4.

[21] P. J. Bentley, "Evolving Fuzzy Detectives: An Investigation into the Evolution of Fuzzy Rules," in *Soft Computing in Industrial Applications*, Y. Suzuki, S. Ovaska, T. Furuhashi, R. Roy, and Y. Dote, Eds., ed: Springer London, 2000, pp. 89-106.

[22] A. Bucchiarone, C. Cappiello, E. Nitto, R. Kazhamiakin, V. Mazza, and M. Pistore, "Design for Adaptation of Service-Based Applications: Main Issues and Requirements," in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. vol. 6275, A. Dan, F. Gittler, and F. Toumani, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 467-476.

[23] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan, "Context is key," *Commun. ACM,* vol. 48, pp. 49-53, 2005.

[24] B. Pernici and S. H. Siadat, "A Fuzzy Service Adaptation Based on QoS Satisfaction," *Advanced Information Systems Engineering,* vol. 6741, p. 14, 2011.

[25] T. M. Nguyen and Q. M. K. Wu, "Dynamic Fuzzy Clustering and Its Application in Motion Segmentation,", *IEEE Transactions on Fuzzy Systems,* vol. PP, pp. 1-1, 2013.

[26] P. Angelov and R. Buswell, "Evolving rule-based models: A tool for intelligent adaptation," in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, 2001, pp. 1062-1067 vol.2.

[27] (2012, 14-06-2013). *The Modum Project*. Available: http://modum-project.eu/

[28] F. Irmert, T. Fischer, and K. Meyer-Wegener, "Runtime Adaptation in a Service-Oriented Component Model," in *Proceedings of the 2008 International Workshop on Software Engineering for Adaptive and Self-Managing Systems* Leipzig, Germany, 2008, p. 8.

[29] K. Wang, M. Dumas, C. Ouyang, and J. Vayssiere, "The Service Adaptation Machine," in *Proceedings of European Conference on Web Services*, Dublin, Ireland, 2008, p. 10.

[30] J. Cámara, C. Canal, J. Cubo, and E. Pimentel, "Dynamic Adaptation Using Contextual Environments," in *Proceedings of the Third International Workshop on Coordination and Adaptation Techniques for Software Entities. WCAT'06*, Nantes, France, 2006, p. 8.

[31] M. Cremene, M. Riveill, and C. Martel, "Towards Unanticipated Dynamic Service Adaptation," in *Proceedings of the Third International Workshop on Coordination and Adaptation Techniques for Software Entities. WCAT'06*, Nantes, France, 2006, p. 10.

[32] C. Baladrón, J. M. Aguiar, B. Carro, L. Calavia, A. Cadenas, and A. Sánchez-Esguevillas, "Framework for Intelligent Service Adaptation to User's Context in Next Generation Networks," *IEEE Communications Magazine,* vol. 50, p. 8, 2012.

[33] D. Miorandi, L. Yamamoto, and P. Dini, "Service Evolution in Bioinspired Communication Systems," *International Transactions on Systems Science and Applications,* vol. 2, p. 9, 2006.

[34] T. Nakano and T. Suda, "Self-organizing Network Services with Evolutionary Adaptation," *IEEE Transactions on Neural Networks,* vol. 16, p. 10, 2005.

[35] M. Treiber, H.-L. Truong, and S. Dustdar, "On Analyzing Evolutionary Changes of Web Services," in *Proceedings of the Service-Oriented Computing – ICSOC 2008 Workshops*. vol. 5472, G. Feuerlicht and W. Lamersdorf, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 284-297.

[36] P. Kaminski, H. Müller, and i. M. Litoiu, "A Design for Adaptive Web Service Evolution," *Proceedings of the 2006 International Workshop on Self-adaptation and Self-managing Systems,* p. 7, 2006.

[37] S. Ponnekanti and A. Fox, "Interoperability Among Independently Evolving Web Services," *Middleware 2004,* p. 21, 2004.

[38] D. Frank, L. Lam, L. Fong, R. Fang, M. Khangaonkar, " in ,, and 2008, "Using an Interface Proxy to Host Versioned Web Services," *IEEE International Conference on Services Computing, 2008. SCC'08,* vol. 2, p. 8, 2008.

[39] V. Andrikopoulos, S. Benbernou, and M. Papazoglou, "Managing the Evolution of Service Specifications," *Advanced Information Systems Engineering,* p. 16, 2008.

[40] F.-L. Li, L. Liu, and J. Mylopoulos, "Software Service Evolution: A Requirements Perspective," *Proceedings of the IEEE 36th International Conference on Computer Software and Applications Workshops,* p. 6, 2012.