DOI: http://dx.doi.org/10.1109/PCI.2011.27

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Elastic Component Characterization with respect to Quality Properties: 
An intuitionistic fuzzy-based approach

George Kakarontzas
Dept. of Informatics
Aristotle University of
Thessaloniki, Greece,
gkakaron@teilar.gr

Vassilis C. Gerogiannis
Dept. of Project Management,
Technological Education Institute
of Larissa, Greece
gerogian@teilar.gr

Ioannis Stamelos &
Panagiotis Katsaros
Dept. of Informatics
Aristotle University of
Thessaloniki, Greece,
{stamelos,katsaros}@csd.auth.gr

Abstract—Component selection based on quality properties is a fuzzy process because measurable component attributes cannot be attributed with certainty to high-level quality properties such as the ones proposed by the ISO/IEC 9126 quality model and other similar models. In addition, measurable component quality attributes can be characterized differently for different application domains (e.g., a total execution time value can be considered very satisfactory for one application domain and extremely unsatisfactory for another). In this paper we demonstrate the usage of an intuitionistic fuzzy approach in selecting components originating from an elastic component repository. Elastic components are the output of a quality-driven process for component development that results in component variants based on quality discrimination. During reuse, utilization of intuitionistic fuzzy sets can be proven an efficient solution to derive a first-level characterization of the available components, by considering not only the uncertainty of the reusers but also their hesitation degree.

Keywords: Elastic components; quality requirements; intuitionistic fuzzy sets; component selection.

I. INTRODUCTION

Component-based software engineering (CBSE) is a software development approach that distinguishes between development for reuse and development with reuse. When we develop components for reuse there is necessarily an attempt to create a component in such a way so that it becomes easy to reuse in many different application contexts. However, it is questionable if we can foresee all different application contexts in advance. Furthermore, even when all development contexts are known in advance, as for example in a Software Product Line with known variants from the phase of scoping [1], it is still challenging to create an all-encompassing component and problems may arise in the future when new unforeseen requirements emerge [2]. In component-development with reuse the problem is to discover and select appropriate components given that the application context is now known. The reuser wants to select a component variant that better addresses her needs. Given that requirements, especially ones concerning quality properties, are often ill-defined and vague, optimizing the selection process is difficult and error-prone.

In previous work ([3], [4]) we have suggested a component development process, based on elastic components, that attempts to mitigate this tension between the unpredictability in component deployment contexts and the flexibility of the evolution of the component. In the current work, we discuss a component selection approach during development with reuse, based on intuitionistic fuzzy sets [5], which assumes an existing elastic component hierarchy. Although fuzzy-based component selection approaches have been proposed (e.g. [6], [7]), our approach also considers the degree of indeterminacy of evaluating quality which is known to be elusive and ill-defined [8]. In addition, our approach differs from these previous works, in that it allows the reuser to interpret existing measurable attributes, which are available from the elastic component repository, in relation to the application requirements. Measurements are not needed to be repeated by the reusers.

Intuitionistic fuzziness is introduced, during component selection for reuse, for four reasons:

a) Measurable component quality attributes, one of the outputs of the elastic component development method, can be characterized differently for different application domains (e.g. a total execution time value, can be considered very satisfactory for one application domain and extremely unsatisfactory for another).

b) Measurable component attributes (e.g. total execution time, memory usage etc.) cannot be attributed with certainty to high-level quality properties (e.g. timing behavior, stability etc.) such as the ones proposed by the ISO/IEC 9126 [9] quality model and other similar models.

c) Concerning (a) and (b) there is also a hesitation aspect involved in the reusers’ assessments. Reusers may be hesitant to some degree to results provided by the component developers’ measurements.

d) Given a potentially large number of components in a component elastic hierarchy, with the same essential functionality but different quality properties, there is a need to efficiently derive a short list of candidate components before preceding to a more precise assessment (for example using search-based techniques such as the one proposed in [10]).

In the rest of the paper, Section II provides an introduction to Elastic components and Intuitionistic fuzzy sets and discusses our approach for component selection based on intuitionistic fuzzy sets and using the elastic component hierarchy. Then in Section III we provide an example of applying our suggested approach for component
selection. Finally, in Section IV we discuss related work and in Section V we provide future research directions and conclusions.

II. BACKGROUND AND APPROACH

A. Elastic components

Elastic components ([3], [4]) are not just a single component but a hierarchy of components with a common root. The children of this hierarchy are variants of the root component which is called pure. The pure component provides a service (e.g., image compression). However, the different variants of this root may provide this service with different quality properties (e.g., improved Total Execution Time or improved Memory Usage). Although variants belonging to different paths in the elastic component hierarchy provide the same service, they may have conflicting quality properties. For example, a component that provides improved Total Execution Time may require increased Memory Usage.

Variants have some functional or quality additions as compared to the pure component. Evolving an elastic component means:

a) Deciding which pure or variant component to extend by choosing the most appropriate node in the hierarchy for the extension.

b) Extending this component with a new variant satisfying the quality or functional goal at hand.

c) Verify the compatibility of the new component with its base component (i.e., verify the new component placement in the elastic component hierarchy).

The verification aspect entails the obligation on behalf of the component developer to verify the improvement of the component under additional functional or quality related properties. As can be seen in Fig.1, the pure component is extended in different directions with variants. Each variant provides the same features as its base but with an improved quality property. If we follow the path from a root component to a variant, the components that belong to this path are refinements of their ancestors.

One problem that arises during component selection in an elastic component hierarchy is that the quality properties of each variant that we improve or impair are not quantitatively determined. For example, the developer may have applied a design pattern or an architectural tactic that is known to have a positive contribution to a quality aspect, and then the reuser verified the improvement by some verification method (e.g., testing or simulation). However, the improvement impact is rather fuzzy (e.g., if Total Execution Time is 10ms is this good, very good or bad?). The actual suitability of the component for a particular application is therefore application dependent, and only the reuser can assess it for her specific requirements at hand.

Another problem is that when reusers want to select a component, they will be interested in finding a component with improved accuracy or stability, which are high-level quality requirements. Low-level measurable attributes that the component developer verified during the elastic component development (e.g., total execution time, memory usage etc.) cannot be related to high-level quality properties (e.g. accuracy, stability) with a precise, unanimously agreed function. The aforementioned problems lead us to investigate the possibility of using intuitionistic fuzzy sets for the selection of elastic components.

B. Short Introduction to Intuitionistic Fuzzy Set (IFS)

An IFS A in a finite set X can be defined as [5]:

\[ A = \{x, \mu_A(x), v_A(x) \mid x \in X\} \]

where:

\[ \mu_A : X \rightarrow [0,1] \]

\[ v_A : X \rightarrow [0,1] \]

\[ 0 \leq \mu_A(x) + v_A(x) \leq 1 \quad \forall x \in X. \]

The functions \( \mu_A(x) \) and \( v_A(x) \) denote respectively the degree of membership and non-membership of \( x \) to \( A \).

For each Intuitionistic Fuzzy Set (IFS) \( A \) in \( X \), \( \pi_A(x) = 1 - \mu_A(x) - v_A(x) \) is called the hesitation degree of whether \( x \) belongs to \( A \). If the hesitation degree is small then knowledge whether \( x \) belongs to \( A \) is more certain, while if it is great then knowledge on that is more uncertain.

Thus, an ordinary fuzzy set can be written as:

\[ \{< x, \mu_A(x), 1 - \mu_A(x) > \mid x \in X\} \]

In the approach, we suggest the use of linguistic terms (Table I) to express uncertainty about the values of the components measurable attributes and the uncertainty about the impact of each component measurable attribute on the required system quality properties. For example, an Intuitionistic Fuzzy Number (IFN) \([0.5, 0.4, 0.1]\) represents membership \( \mu = 0.5 \), non-membership \( v = 0.4 \) and hesitation degree \( \pi = 0.1 \). This IFN may characterize the total execution time of a component during reuse as medium. 0.5 stands for agreement to this characterization, 0.4 with disagreement and 0.1 as hesitation between these two extremes. As we already mentioned, such characterizations can only be expressed by the reuser during component reuse, based on the application domain.
C. Approach

The approach that we propose complements the elastic component building method and can be described with the following steps:

During component development (development for reuse):

a) The component developer constructs a component variant hierarchy based on modifications of a pure component.

b) Each new variant is exercised by an appropriate validation method (e.g., testing, simulation etc.) to derive values for the measurable attributes that describe the quality improvement. In addition, for each variant, the verification artifacts of its base are also exercised, to determine that the variant is indeed a refinement of its base.

During application development (development with reuse):

c) According to the application domain, the reuser characterizes, by using linguistic terms (Table I), the measurable attributes of interest for the components and measurements produced in steps (a)-(b).

d) The reuser also quantifies the impact of the different measurable attributes to the quality properties of the application (Table I).

e) To derive efficiently a characterization of available components in relation to all quality properties and attributes, the reuser calculates reliable distance measures provided by the intuitionistic fuzzy domain (described in the following Section).

III. CASE STUDY

To allow comparison of our results with other approaches we will use the components discussed in [11]. In this work the authors compare image compression components with the Analytic Hierarchy Process (AHP) using important measurable attributes. The attributes used are the Total Execution Time (TET), Memory Usage (MU), Compression Ratio (CR) and Root Mean Square Error (RMSE). For these attributes the authors carried measurements using images of various sizes and types. The output of their work provides an average and a standard deviation for each component and each measurable attribute. The components used implement the Arithmetic Encoding (AREC), Huffman coding (HUFF), Burrows-Wheeler Transform (BWT), Fractal Image Encoding (FRAC), and Embedded Zero-Tree Wavelet Encoder (EZW) compression techniques. In the following, we use these results to derive Intuitionistic Fuzzy Numbers (IFNs) based on the reuser application domain and needs. These IFNs are depicted in Table II. The reuser interprets available crisp measurements and provides subjective judgments on the degree that each component satisfies the measurable attributes. For example, available measurements in the form of “<average, std>” of TET (in s) for (HUFF, AREC, BWT, FRAC, EZW) are (<111.3, 132.4>, <180.3, 237.3>, <473.3, 371.1>, <89.5, 73.3>, <380.4, 485.4>), where each component has been executed multiple times for a set of images. Based on these measurements, and according to application requirements, the reuser assigns the linguistic values (Positive, Positive, Very Negative, Very Positive, Very Negative) which are encoded in the respective IFNs of Table I. Analogous interpretation takes place for the rest of the measurable attributes.

The next step in the selection process is to evaluate the impact of each measurable attribute to the high-level quality properties of interest. In the case study, these high-level quality properties are Time Behavior, Resource Utilization, Accuracy, Testability and Stability. We assume that these quality properties are important for the application domain of the reuser. We emphasize that depending on the application domain, quality properties used for this step may be any of the numerous properties of ISO/IEC 9126 [9] or other suitable quality models. The reuser assigns linguistic valuations on the degree that each measurable attribute impacts positively the corresponding quality property. In this way, the application requirements are quantified. For the case study these valuations are presented in Table III. The values in Table III were derived by assigning the IFNs of Table I to each measurable attribute and quality property combination.

Given the two tables (Table II and Table III) we are now able to calculate the distances for each component from the considered set of quality properties. In particular, distance measures can be used to evaluate the suitability of each candidate component in supporting the related properties. As distance measures we have used the intuitionistic Hamming distance and the intuitionistic Euclidean distance. Both distances have been proved to be reliable distance measures since they take into account not only membership and non-membership, but also the hesitation part of IFNs [12]. Euclidean distance is useful when the Hamming distance results in a tie.

The calculation of Hamming distances is performed by using equation (1). For each component \( c_i \) (\( i=1..5 \)), in equation (1), the component’s measurable attributes values are \( a_j \) (\( j=1..4 \)) and the quality properties are \( q_k \) (\( k=1..5 \)). By observing Table IV, the most suitable candidates for testability, resource utilization, and stability are HUFF, AREC and BWT, respectively. We observe also that FRAC is equally suitable to support time behavior and testability, where EZW is also a good candidate for supporting stability.

### TABLE I. LINGUISTIC TERMS USED DURING EVALUATIONS

<table>
<thead>
<tr>
<th>Linguistic Terms for rating component measurable attributes</th>
<th>Intuitionistic Fuzzy Numbers (IFNs)</th>
<th>Linguistic Terms for rating of impact measurable attributes have on quality properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Positive (VP)</td>
<td>( \mu ) 0.9 ( v ) 0.1 ( \pi ) 0.0</td>
<td>Very Strong Impact (VSI)</td>
</tr>
<tr>
<td>Positive (P)</td>
<td>( \mu ) 0.7 ( v ) 0.2 ( \pi ) 0.1</td>
<td>Strong Impact (SI)</td>
</tr>
<tr>
<td>Medium (M)</td>
<td>( \mu ) 0.5 ( v ) 0.4 ( \pi ) 0.1</td>
<td>Medium Impact (MI)</td>
</tr>
<tr>
<td>Negative (N)</td>
<td>( \mu ) 0.3 ( v ) 0.6 ( \pi ) 0.1</td>
<td>Low Impact (LI)</td>
</tr>
<tr>
<td>Very Negative (VN)</td>
<td>( \mu ) 0.1 ( v ) 0.9 ( \pi ) 0.0</td>
<td>Very Low Impact (VLI)</td>
</tr>
</tbody>
</table>
To derive a more concrete indication, we have calculated the Euclidean distances by using equation (2). The results presented in Table V show again that HUFF is a good candidate for testability, AREC for resource utilization and BWT for stability, as before. Now we can conclude that FRAC performs better in relation to Testability than its other quality properties, so the tie is resolved. EZW also performs better in relation to stability.

This is a small case study suitable for demonstrating the approach. In real-world scenarios it is quite possible that the number of components, measurable attributes and quality properties will be quite large. However, the effectiveness of the calculations can be proven useful for the reuser to derive a short list of candidate components first and then apply a more exhaustive examination of their suitability.

IV. RELATED WORK

Our approach in evaluating components based on intuitionistic fuzzy distances is inspired by the work in [13] which uses these measures for the medical diagnosis domain. The advantage of this technique is that there is no need to calculate any new composition of intuitionistic fuzzy relations (e.g., Max-Min-Max) and thus, there is no need to apply an inference engine to arrive to conclusions about component suitability. Actually, this is the main difference of our approach with others such as the one presented in [14], where the authors use inference rules for selecting and retrieving components. Compared to other component selection methods using fuzzy sets (e.g. [6], [7]), our approach also differs in that it distinguishes clearly the phases of component development and component reuse. Thus, we adopt a selection perspective that is only related to the component reuse phase which can be independent from the component development phase [15]. The reuser needs only to interpret available values of measurable attributes from the elastic component repository. Furthermore, reducing the component set, by applying the proposed approach, can be applied prior to an exhaustive search-based component selection process ([10]).

Our approach in constructing the elastic component hierarchy can be compared to agile methods of program development and Software Product Line (SPL) approaches to software development because (a) it uses small development steps followed by verification activities (which is similar to some extent to the Test-Driven Development (TDD) approach of agile methods) and (b) variants of base components which add new quality or functional features (which is inspired by the SPL approach). In fact, it can be considered as an Agile Product Line Engineering (APLE) approach, a recent research area attempting to merge the benefits of agile and SPL approaches [16], [17].

\[
H(a(c_i), q_k) = \frac{1}{2 \cdot 4} \sum_{j=1}^{4} (\mu_j(c_i) - \mu_j(q_k))^2 + (v_j(c_i) - v_j(q_k))^2 + (\tau_j(c_i) - \tau_j(q_k))^2
\]

(1)

\[
E(a(c_i), q_k) = \sqrt{\frac{1}{2 \cdot 4} \sum_{j=1}^{4} (\mu_j(c_i) - \mu_j(q_k))^2 + (v_j(c_i) - v_j(q_k))^2 + (\tau_j(c_i) - \tau_j(q_k))^2}
\]

(2)

We have explored this interplay between agile methods of software development and SPLs in our previous works [18], [19].

V. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this work we presented an approach to software component development (i.e., the elastic component development approach) which is quality-driven and combines characteristics of agile and SPL-based approaches. Elastic components variants are quality transforming components of their base components. To allow effective and efficient application-dependent searching and retrieval we augmented the elastic component development method with an intuitionistic fuzzy-based approach to component selection.

The approach raises several issues that could spark further research. For example, an interesting idea could be to validate the approach applicability in addressing the selection of Open Source Software components. In addition, treating more with uncertainties in component selection would further strengthen the proposed approach in deriving more precise results. Therefore, we have plans to examine the utilization of more powerful methods, such as the interval-valued intuitionistic fuzzy sets [20].

REFERENCES


<table>
<thead>
<tr>
<th>TABLE II.</th>
<th>VALUES OF COMPONENT MEASURABLE ATTRIBUTES PER COMPONENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TET</td>
</tr>
<tr>
<td>HUFF</td>
<td>0.7</td>
</tr>
<tr>
<td>AREC</td>
<td>0.7</td>
</tr>
<tr>
<td>BWT</td>
<td>0.1</td>
</tr>
<tr>
<td>FRAC</td>
<td>0.9</td>
</tr>
<tr>
<td>EZW</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III.</th>
<th>VALUES OF IMPACT OF MEASURABLE ATTRIBUTES TO COMPONENT QUALITY PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TIME BEHAVIOR</td>
</tr>
<tr>
<td>TET</td>
<td>0.9</td>
</tr>
<tr>
<td>MU</td>
<td>0.7</td>
</tr>
<tr>
<td>CR</td>
<td>0.3</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV.</th>
<th>HAMMING DISTANCES OF EACH COMPONENT FROM THE QUALITY ATTRIBUTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME BEHAVIOR</td>
<td>RESOURCE UTILIZATION</td>
</tr>
<tr>
<td>HUFF</td>
<td>0.33</td>
</tr>
<tr>
<td>AREC</td>
<td>0.28</td>
</tr>
<tr>
<td>BWT</td>
<td>0.45</td>
</tr>
<tr>
<td>FRAC</td>
<td>0.33</td>
</tr>
<tr>
<td>EZW</td>
<td>0.38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE V.</th>
<th>EUCLIDEAN DISTANCES OF EACH COMPONENT FROM THE QUALITY ATTRIBUTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME BEHAVIOR</td>
<td>RESOURCE UTILIZATION</td>
</tr>
<tr>
<td>HUFF</td>
<td>0.34</td>
</tr>
<tr>
<td>AREC</td>
<td>0.32</td>
</tr>
<tr>
<td>BWT</td>
<td>0.53</td>
</tr>
<tr>
<td>FRAC</td>
<td>0.37</td>
</tr>
<tr>
<td>EZW</td>
<td>0.52</td>
</tr>
</tbody>
</table>