A Decision Support System for Software Architecture-Style Selection

Shahrouz Moaven  
Department of Computer Engineering  
Sharif University of Technology  
moaven@ce.sharif.edu

Jafar Habibi  
Department of Computer Engineering  
Sharif University of Technology  
jhabibi@sharif.edu

Hamed Ahmadi  
Department of Computer Engineering  
Qazvin Azad University  
hamed_ahmadi@qazviniau.ac.ir

Ali Kamandi  
Department of Computer Engineering  
Sharif University of Technology  
kamandi@ce.sharif.edu

Abstract

Due to the enlargement and complexity of software systems and the need for maintenance and update, success of systems depends strongly on their architecture. Software architecture has been a key element in software development process in two past decades. Therefore, choosing the correct architecture is a critical issue in software engineering domain, with respect to the extremely extension of architecture-driven designs. Moreover, software architecture selection is a multi-criteria decision-making problem in which different goals and objectives should be considered.

In this paper, a Decision Support System (DSS) has been designed which provides software architects with more precise and suitable decisions in architecture styles selection. The DSS uses fuzzy inference to support decisions of software architects and exploits properties of styles in the best way while making decisions.

1. Introduction

During the software system development lifecycle, choosing a suitable architecture for the system and its homogeneous subsystems is an essential parameter [1]. Making use of architecture styles is one of the ways to design software systems and guarantee the satisfaction of their quality attributes [2].

Many factors/criteria affect the architecture styles selection which makes it a multi-criteria decision making problem. Decision making problem is the process of finding the best option from all of the feasible alternatives. In a multi-criteria decision making problem, the multiplicity of criteria for judging the alternatives is pervasive.

There are some quality attributes for each architecture style that may have good or bad effect(s) on different domains. Although some attributes have been listed for each style in different texts, but we cannot understand the extent to what benefits and disadvantages of quality and quantity attributes of architecture are considered [3]; therefore, comparing capabilities and benefits of software architecture is a somehow difficult task. Moreover, capabilities of software architecture styles may have been listed with respect to a special domain; consequently, software architectures should be refined and completed in accordance with architects' experiments. As a result, deciding about the architecture style(s) that should be selected depends directly on the realization/intuition of the system architect.

In order to provide system architects with more precise and suitable results, a Decision Support System has been proposed in this paper which tries to support the decision making process by considering different related criteria and suggesting suitable solutions. It should work as a complement to intuitive judgment while selecting architecture.

In order to solve multi-criteria decision making problems, there is a need for solutions which aggregate different criteria in order to come into one overall result. In this way, some suitable solutions have been proposed in fuzzy logic for criteria aggregation which are able to model interaction between criteria in a flexible way [4]. Thanks to the flexibility of using fuzzy tools, which will be described in continue, provided results could be considered as the most suitable ones for the domain. Therefore, the designed decision support system would reduce the level to which architects should deal with complexities of selecting architecture styles.

The remainder of this paper is organized as follows. Importance of software architecture and selecting a
suitable architecture style is discussed in section 2. Section 3 introduces the basic concepts of decision support systems and making use of them in software design, in addition to the importance of fuzzy logic in DSS. Some basic definitions for solving a multi-criteria decision making problem by fuzzy Choquet integral is defined in section 4. The proposed fuzzy-based DSS is presented in next section. Section 6 contains an example followed by conclusion in the final section.

2. Software Architecture

Software architecture is system structure(s) which specifies software components, their specific properties and relations among them [5]. Software architecture is known as a framework to fulfill requirements and a managerial and technical basis for cost estimation and process management. Architecture is a criterion for measuring concurrency and dependency of components that enables us to exploit it as a guideline whole over the project [2, 7]. In short, architecture is the conceptual glue that holds every phase of the project together [6].

Due to the extreme increase in the complexity of software systems, making use of software architecture to address it has become of importance. In other words, increasing the understandability and manageability of the system and the system’s development process is an important goal of software architecture [1]. By defining system architecture, we become able to know and exploit design vocabularies, permitted structural components, computational models, and advantages and disadvantages of different methods and techniques. A crucial issue in success of software systems is the high quality of their architecture. This matter comes to be of importance when both systems and their software architecture are complex. Therefore, a high quality architecture that can help meeting the architecture requirements will provide more success for the system [1].

2.1. Architecture Styles

An architecture style provides us with a glossary of component and connector types and also a set of rules about the way of their combination [2]. One (or more) semantic model(s) may exist to specify the general properties of a system regarding the specifications of its elements. Therefore, the system can be described as a style or a composition of styles [5]. Styles can loosely be seen as ways to group components, where the components are related to each other through structure. It provides us with exploitation of suitable structural and evolution patterns and facilitates component, connector, and process re-use [5]. Therefore, we can take advantage of reusability of these patterns.

Architecture styles classified by Shaw and Garlan in 1994 and have been applied in many projects till now. Therefore, a set of proved qualitative and quantitative properties are defined for these reusable structures that lead to simplicity and trustworthiness in choosing and using them in new projects and special domains [5, 7]. In accordance with specified rules and frameworks of architecture styles, use of them is simple and prevents architecture users from being involved in processes and complexities of an ad-hoc architecture design. We should mention the point that some changes should be performed in architecture styles or a combination of them should be used to gain better adaptation with problem domain and increase the architecture performance.

2.2. Selecting Architecture

One of the important parts in software design process is the selection of software architecture. Choosing the most suitable architecture style(s) among existing ones can help us in satisfying functional and especially non-functional requirements correctly and precisely [7].

As we should consider different goals and objectives in architecture styles selection, such as functional and non-functional requirements, architect’s priorities and domain commitments, selecting architecture styles is categorized in multi-criteria decision-making problems. The architect should select architecture styles in a way that they satisfy all criteria related to the problem in the best way. Calculating the effectiveness and importance of each requirement and aggregating them are some important issues that turn the architecture style selection into a complex problem. Therefore, we need a powerful and precise decision support system to help and support system architects. For selecting architecture styles, evaluation methods and techniques [8] are usually used; but these methods do not pay attention to the abilities and capabilities of styles. In continue, a decision support system is presented (in section 5) that includes all useful information and uses suitable tools to make decision.

3. Decision Support Systems

Due to the extremely high attention to the computer-based information systems, making use of Decision Support Systems to support and improve decision making has become of importance. Decision
making problem is the process of finding the best option from all of the feasible alternatives. DSS is a computer-based system which supports a decision, in any way. DSS will essentially solve or give options for solving a given problem. A DSS provides support for all phases of the (semi-structured and unstructured) decision making process and a variety of decision making processes. It should be easy to use meanwhile providing support for users at all levels to make decision [9].

Decision support systems can be classified in several ways. One well-known classification is to put them into six frameworks [9]: 1) text-oriented which emphasizes on creating, revising and reviewing documents; 2) database-oriented in which the database organization is of importance and the emphasize is most on query capabilities and generating strong reports; 3) spreadsheet-oriented which allows the user to develop models to execute DSS analysis; 4) solver-oriented DSS which use solver for solving a particular type of problem; 5) rule-oriented DSS in which the knowledge component, which often is an expert system, includes procedural and inferential rules; 6) compound DSS which is a combination of two or more of classifications mentioned above.

In order to construct a DSS, it is important to know the cognitive abilities of the user(s) of it (i.e. decision maker) and the way they arrive at a decision. Furthermore, the purpose of the DSS, external sources, internal data files, and the major processes in the DSS should be determined before starting to design [11].

3.1. Fuzzy Decision Support System

In many cases, due to the complexity of the problems, determining optimal solutions in decision problems cannot be done by using a single criterion or a single objective function [12]. For human decision makers, it is difficult and time consuming to take into account all criteria related to a problem at once; moreover it is more difficult to consider interaction among them and finally make the best decision. Dealing with interacting criteria is a kind of difficult issue which people usually tend to avoid the problem by constructing independent (or supposed to be so) criteria [4, 13].

In order to make decision in environments in which we have ambiguity and uncertainty and we cannot distinguish between the concepts, making use of fuzzy logic [15] could be considered as one of the solutions. Making use of fuzzy logic in decision support systems allows linguistic terms to be converted into numerical data. Moreover, it can be combined with other techniques to achieve better performance [9].

3.2. Decision Support in Software Design

As mentioned, making use of decision support systems has had an increase in recent decade. Statistics show that this increase has been in making use of decision support systems to support high-level managerial decisions, while use of them in software development process is negligible [14]; because software engineers did not feel any need to them in their own domain. But due to the fast increase in complexity of systems on one hand and competitive business on the other hand, software development has become a challenging process in which the role of decision makers is so important that could completely change the destiny of the project. Therefore a need for decision support systems in this domain is appreciated. This DSS should be able to deal with multi-criteria problems in order to be applied in software architecture selection domain.

Being able to use or aggregate different techniques with DSS, they have become able to be applied in finer granular levels. Constructing and developing DSS and making use of them as a middle-level tool in software development process could be a suitable approach to overcome complexities of making decision in this field. In continues we will represent a decision support system to support decisions that are made in software architecture selection. As we use a type of fuzzy integral in the decision support system, at first, some basics about fuzzy logic and the definition of Choquet integral is reviewed in the next section before describing the overall structure of the decision support system.

4. Fuzzy Choquet Integral

As mentioned before, we face with a multiple criteria decision making problem in which, criteria may interact in a negative or positive way. In the former case, there is a kind of redundancy between two criteria while in the latter a kind of synergy exists among them. Taking advantage of subadditivity of the fuzzy measures can model negative interaction among criteria. If they interact positively, taking advantage of superadditivity implies synergy. After finding the importance level of sets of criteria, a solution to aggregate them and find a final result to help decision making process is needed.

Several approaches have been proposed to be used in multi-criteria decision making processes. A common
important feature of fuzzy integral is the ability of representing a range of interaction between criteria: from redundancy to synergy [4, 13]. In other words, we can consider both negative and positive interactions between different criteria.

But first some basic definitions should be explained.

1. In a multi-criteria decision making problem, there exists a set of alternatives \( \Omega \) among which the decision maker must choose. Each alternative is associated with a set of criteria \( C \) which represents what the decision maker will get by choosing this alternative [4, 13].

\[
\Omega = \{a_1, a_2, \ldots, a_n\},
\]

\[
C = \{c_1, c_2, \ldots, c_n\}
\]

2. Let \( P(C) \) be the power set of \( C \). A fuzzy measure on \( (C, P(C)) \) is a set function \( \mu : P(C) \rightarrow [0,1] \) such that

i. \( \mu(\emptyset) = 0 \), \( \mu(c_1, \ldots, c_n) = 1 \)

ii. if \( A, B \in P(C) \) and \( A \subset B \) then

\[\mu(A) \leq \mu(B)\]

In this context, \( \mu(A) \) is a fuzzy measure which represents the degree of importance of the set of criteria \( A \). Therefore, we should define \( 2^n \) coefficients corresponding to the \( 2^n \) subsets of \( C \).

3. The fuzzy measure \( \mu \) is superadditive if \( \mu(A \cup B) \geq \mu(A) + \mu(B) \).

4. The fuzzy measure \( \mu \) is subadditive if \( \mu(A \cup B) \leq \mu(A) + \mu(B) \).

5. The Choquet integral [4, 13] of an application \( f : X \rightarrow [0, +\infty) \), with respect to the fuzzy measure \( \mu \) and taking into consideration that \( X \) is a finite set, is defined by:

\[
C_\mu(f(x_1), \ldots, f(x_n)) = \sum_{i=1}^{n} (f(x_{(i)}) - f(x_{(i-1)})) \mu(A_{(i)}) \quad (1)
\]

Where \( f(x_{(0)}) = 0 \), \( f(x_{(1)}) \leq \ldots \leq f(x_{(n)}) \)

and \( A_{(i)} = \{x_{(i)}, \ldots, x_{(n)}\} \).

The way of using this formula to compute, is shown in the example represented in section 6.

5. Fuzzy DSS for supporting Architecture Styles Selection

In order to select architecture styles correctly and precisely, as discussed previously, all existing information related to the project should be considered; therefore, it is a multi-criteria decision making problem. Furthermore, this information may interact with each other in some cases, which makes it difficult to select the best architecture style. In this way, making use of a decision support system which is able to consider not only different criteria but also the interaction among them is a solution to solve the problem in a more disciplined way which could provide software architects with most suitable results for the domain.

As mentioned in section 3, in order to construct a decision support system, cognitive abilities of the users of DSS should first be understood. Nowadays, because of the lack of such systems, style selection depends on expert architects. By having a decision support system, one can imagine that the users of this decision support system are software engineers that have enough knowledge about quality attributes in different domains and the degree of redundancy or synergy among them.

But, in order to take advantage of all useful information that could help in performing precise decisions a decision support system was designed which is indicated in Fig. 1. As it is illustrated, it has been included of six essential components which make the DSS able to store, extract, and add all necessary information which is helpful to make decision. These components are:

- Domain repository
- Styles repository
- Rule base
- Tools
- Decision maker
- User interface

Duties and responsibilities of each component in addition to what they provide for the system in order to make decision is demonstrated below.

- **Domain repository** - the information about domain of the project at hand can be used as a support for the architect’s decision about the importance of each criterion (i.e. when he is going to provide priority of quality attributes which must be satisfied). Furthermore, this information is used to extract rules which are needed to make decision. The information has been stored in the domain repository and can be retrieved when requested. In other words, domain repository contains the information which can explain the extent to what each quality attribute is of importance with respect to different domains. In the domain repository, similarity of domains could be determined in order to obtain better and more precise decisions. Furthermore, this information could be updated after each decision making process with respect to new results in various domains.
Fig. 1 Decision Support System Architecture to Architecture-Style Selection

**Styles repository**- this repository should contain all styles and patterns which have been used in different projects frequently; the information about the level to which different quality attributes are satisfied by different styles has been stored in it. This information has been stored in the style repository experimentally with respect to the past best practices. In fact each criterion has been assigned a (predefined) number for each architecture style that will be used to help decision making process.

The style repository generally includes mechanisms of storage, search, and update. It maintains styles and patterns plus their categories, the relation among them and the experimental information about their usage. This repository provides a method for searching and extracting information too; furthermore, it can be integrated by tools for searching or storing information. Style repository can be updated after each decision making process; the changing should be performed by expert architect with enough attention to special situation of the project.

**Rule Base**- in order to make decisions, we need some rules which indicate the interaction among quality attributes in an architecture style with respect to the domain. In other words, the rules determine the level of importance of all possible sets of criteria. These rules are kept in the rule base and are extracted from architecture styles repository and domain repository. Therefore, as these two repositories could be updated after making decisions, the maturity of rules would increase by time which will lead to more precise decisions. The general form of rules is:

\[ \text{IF } Q, \text{ THEN } \mu(Q_i) = x_{Q_i} \]

where \( Q_i = \{ q_i | q_i \subseteq P(Q) \}; i = 1, \ldots, 2^n \).

Therefore, for a set of \( n \) criteria, \( 2^n \) rules will exist for each architecture style and will cover all the aspects of the relationships, including synergy and redundancy, among different sets of criteria with respect to the problem. In cases of synergy or redundancy, we should take advantage of superadditivity or subadditivity of fuzzy measures, respectively.

**Tools**- in order to aggregate different criteria related to the problem, we make use of fuzzy integral which was introduced in section 4.

If we want to perform a map between definitions explained in section 4 and architecture styles selection problem, the problem can be defined as: There exists a set of architecture styles \( S \) among which the architect must choose. Each architecture style is associated with a set of quality attributes \( Q \) which represents what the architect will get by choosing this architecture style.

The level of importance of each quality attribute is represented by \( \mu(q_i) \). If two quality attributes are redundant (i.e. negative interaction) the importance level of satisfying both of them is less than sum of importance levels of them; therefore, the subadditivity property of fuzzy measures can model redundancy. If the interaction between two quality attributes is positive (i.e. synergy) the importance level of satisfying both of them is more than sum of importance levels of them; therefore, the superadditivity property of fuzzy measures can model synergy. Furthermore, by taking (1) into consideration, we can come to the conclusion that in this formula \( f(x_i) \) must be replaced with the level of satisfaction of quality attributes by architecture styles. It will become clearer in the example.

When the necessary information is gathered and importance levels of criteria are obtained, the DSS will apply fuzzy Choquet integral for criteria aggregation. The overall process of aggregating criteria is illustrated in Fig. 2. The rules, which were extracted from architecture styles repository and domain repository, and the priorities, which were defined by DSS user, are inputs of the tool. Fuzzy tool will use architect's priorities to obtain importance levels of sets of criteria.
from rules. The fuzzy tool uses these importance levels as coefficients in Choquet integral and applies it on each architecture style.

After applying Choquet integral on all architecture styles which exist in the styles repository, the extent to what each style could satisfy requirements of the project at hand will obtain.

Nevertheless, as an open working area for future research, we will try to use some tools which are more adaptable with the matter and are more precise.

**Decision maker**- the responsibility of this component is, in fact, receiving and sending information from and to all components of the DSS. This component, as an "information propagator" has a duty to receive priorities which architect enters the system via user interface and provide them as an input for the tool which exists in the DSS to make use of it. In this DSS, the priorities are considered as inputs of the fuzzy tool for criteria aggregation. After getting answers which are provided by applying fuzzy integral, and making use of other supplementary information, which could be provided by human agent or useful application tools, the decision maker will determine that an architecture style or a composition of them could be selected for the problem at hand. The results of decision making are sent to the user interface for representation. Therefore, system will suggest them to the architect in order to choose the best one by his experience.

Another responsibility of this part is to recognize new compositions which have been developed during the decision process and to add them into the styles repository. As mentioned, in order to cover the problem domain, it might be needed to use some architecture styles together. In this case, the expert architect will update information repositories by getting feedback from results of the systems which were implemented in the platform of previously selected architectures. In continue, we will describe update occasions.

This part includes an internal human agent who is also the internal expert architect who Make decisions about the updates which will be mentioned in subsection 5.1. The human agent should determine whether the style repository or domain repository should be updated or not. This decision is taken by using the new results which is received from implementation scope.

**User interface**- as mentioned, the user of DSS enters some information to the system with respect to his experiences and information of the domain repository. This information shows the extent to what each quality attribute is of importance in the project at hand and provides the priority of the quality attributes which must be satisfied. The user interface is responsible for receiving this information from user; the received information is entered to the decision maker. Moreover, representing necessary information for the user (information received from decision maker or retrieved from domain repository) is the responsibility of user interface. The suggested architecture style(s) is/are represented by user interface as a suggestion to the architect by their priority. The architect could select the suitable architecture style(s) among suggested one(s) with respect to his knowledge about the problem.

User interface category is a prevalent debate in software engineering which has some exclusive considerations. Discussing about user interface requires more space and demands special work to make the DSS capable of interacting better with its users.

### 5.1. Updating DSS

In order to improve maintainability of the designed DSS, capabilities of updating both information and tools are considered. An important capability of the proposed decision support system is its ability to update which makes the DSS able to use past experiences in the future. As mentioned, update can occur in some components of the DSS.

Domain repository is a component that could be updated after each decision making process whether the domain (or a similar domain) is not in the repository or obtaining different documented results about domain information.

Architecture styles repository is the other component which could be updated whether after each decision making or after implementing the system and obtaining its results. In other words, it could happen whether the used architecture style does not exist in the styles repository (in case of using a composition of architecture styles) or after acquiring the results which are obtained from implementing the system in the platform of selected architecture. Controlling and
managing these updates are performed in the decision maker part under the supervision of expert architect.

Another component which is affected by updates is the rule base. As rules are extracted from styles repository and domain repository, they would be changed when repositories are updated. We can take advantage of association rule mining techniques in order to extract rules from information which are stored in the repositories. It is highly unlikely that the rules would be changed after a long period of usage; therefore, the expertise level of this DSS will increase after each usage.

The explained process is generally separate from the components which are used in the structure; therefore, we can simply enter a new component or replace current components with new ones in case of improving and obtaining better results (e.g. complete or extend the tool).

6. Case Study (Example)

In order to clarify the way of making decision by making use of fuzzy integral in this DSS, an example is represented briefly. One should mention the point that this example is nominal and just tries to show the working process of decision support system.

As an example, imagine we have three criteria named performance (P), functionality (F), and maintenance (M). There are three styles in the styles repository: pipes-and-filters, layered, blackboard. The level of satisfying each quality attribute by means of different styles has been represented in table 1.

<table>
<thead>
<tr>
<th>Architecture-styles</th>
<th>Performance</th>
<th>Functionality</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pies-&amp;-Filters</td>
<td>13</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Layered</td>
<td>1</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>Blackboard</td>
<td>8</td>
<td>15</td>
<td>8</td>
</tr>
</tbody>
</table>

Moreover, imagine that according to the architect, the level of importance of the quality attributes performance, functionality, and maintenance in the project at hand is 4, 3 and 5 respectively.

We imagine that satisfying performance and maintenance concurrently, provides us with double benefit (i.e. they interaction type is synergy) and therefore we can use superadditivity property. Furthermore, in case of functionality and performance there exists a kind of redundancy and therefore we can take advantage of subadditivity.

By using the Rule Base, importance levels can be created as below:

\[
\mu(P) = 0.4, \quad \mu(F) = 0.3, \quad \mu(M) = 0.5; \quad \text{due to the point that maintenance is more important and functionality is less important quality attributes. Moreover, note that the initial ratios (4, 3, 5) are kept unchanged.}
\]

For sets containing two quality attributes between different criteria we have:

\[
\begin{align*}
\mu(F, M) &= 0.8, \\
\mu(P, M) &= 0.98, \\
\mu(F, P) &= 0.5.
\end{align*}
\]

The considerable point is that the weight which is assigned to the set \{P, M\} should be greater than the sum of individual weights. It should be less for the set \{F, P\}.

Finally according to the definition 2 we have:

\[
\mu(P, F, M) = 1
\]

Table 2 represents the obtained values after applying (1) to these input variables.

After this, results will be sent to the decision maker which will combine the incoming results with additional information such as domain information or expert architect idea; consequently, one or more architecture styles will be determined for the architect.

7. Conclusion

In order to satisfying the quality of software systems, making use of suitable system architecture is propounded as a key element. Satisfying quality attributes is one of the important issues in design or selection of true architecture for systems. Furthermore, not only vast number of criteria should be considered,
but also, in order to get best result, interaction among them should be considered too. In this way, making use of a computerized system that can solve the problem and suggest best solutions is of high concern.

In this paper, a decision support system was presented which makes use of fuzzy concept to represent concepts of quality attributes more precisely and efficiently while considering interaction among them. We constructed a DSS based on knowledge-base which has the ability of updating its knowledge and provides the system architect with suitable choices to select among them. With respect to knowledge base and exploiting expertise of people that work in this domain (e.g. expert architect) some rules have been extracted that can help to surfing the style repository and offering a style or combination of some styles.

The capability of updating after each decision making can increase the precision of future decisions by exploiting previous experiments and extending rule base by new rules. This work was the first version of researching in construction of DSS for architecture selection. We decide to enhance the architecture and process execution for improving the quality of our DSS in future.

8. References