Dealing with Imprecise Quality Factors in Software Design

Joost Noppen, Pim van den Broek & Mehmet Aksit
Trese Group, Dept. of Computer Science
P.O. Box 217, 7500 AE Enschede
The Netherlands
{ noppen | pimvdb | aksit }@cs.utwente.nl

Abstract – During the design of a software system impreciseness can manifest itself in for instance the requirements or performance estimations. While it is common to eliminate the impreciseness by information that can not be justified, it is better to model the impreciseness since it is the most accurate description that is available at the current point in time. In this paper we present an approach, which allows the explicit specification of quality estimations and quality requirements including the imprecise nature. In this approach the impreciseness is modeled and addressed using representations from probability theory and fuzzy set theory.

1. Introduction

During the design of a software system a software engineer is faced with many choices. Most of these choices involve the identification of a solution for a problem from a given set of alternatives [1]. The alternative solutions generally only differ with respect to quality attributes such as performance or adaptability, therefore the alternative should be selected which satisfies the quality requirements best. However, it is very difficult to focus on multiple quality concerns simultaneously during the design of a software system. Quality attributes are therefore mostly considered in an ad-hoc manner, and improved to meet requirements at a later stage in the design process, e.g. during integration or testing.

When a design should be selected from a number of alternatives, this is mostly done by comparing the quality attributes that are considered relevant at the current point in time. However, this evaluation can only be done accurately after a software system has been implemented. The measurements on the completed system that are done based on metrics represent the actual behavior of the system. However, the choice for a design alternative is not taken after the completion of a system, but rather at earlier phases of the design process. The earlier a decision should be taken in the design process, the more difficult it is to determine the quality behavior of the resulting system. Therefore estimations are made, which replace the measurements that can not be performed at that point in time.

In traditional methods such as SAAM [2] the trade-off between multiple design alternatives, such like architectural styles or design patterns, is done based on these metrics and estimations. However, even while estimations have an intrinsically imprecise character, in traditional methods they are interpreted in the same manner as precise measurements on a completed software system. The variance that is possible due to the impreciseness is not addressed accurately and this can lead to faulty evaluation of design alternatives.

In this paper we present an approach with which it is possible to evaluate design alternatives with respect to their respective quality restrictions. The approach specifically focuses on supporting impreciseness when expressing non-functional requirements and/or estimations. Non-functional requirements can be expressed in a complete and unambiguous manner (crisp), or in a vague manner.

For the first no additions are needed to existing methods, whereas the second is supported by the framework by using fuzzy sets. For estimations, in addition to the other two types, also probabilistic descriptions are allowed as a means to describe impreciseness. Note that the use of fuzzy sets is a very generic and configurable approach. For specific cases a more specialized fuzzy model can be applied such as for instance fuzzy booleans [6]. In figure 1 the relationships between the types of impreciseness are depicted. In the figure each combination is represented by a tuple (x, y). In this tuple x represents the type of impreciseness in the requirements (being either crisp or fuzzy) and y the estimations (being either crisp, probabilistic, fuzzy or fuzzy probabilistic).
In figure 1 the most complex combination fuzzy requirements & fuzzy probabilistic estimations \((f, fp)\) is located at the bottom of the picture. By removing impreciseness in either the requirements or the estimations, one of the situations on a more simple layer attained. This could be crisp requirements & fuzzy probabilistic estimations \((c, fp)\) or fuzzy requirements & probabilistic estimations \((f, p)\). In addition it is also possible to specify requirements and estimations using fuzzy requirements & fuzzy estimations \((f, f)\). By removing impreciseness from this level, the combinations of a higher level are reached. These could be either crisp requirements & probabilistic estimations \((c, p)\), crisp requirements & fuzzy estimations \((c, f)\) or fuzzy requirements & crisp estimations \((f, c)\). When from this layer the remaining impreciseness is removed you will arrive at the top layer, consisting of crisp requirements & crisp estimations \((c, c)\). Note that current approaches acknowledge only \((c, c)\) and \((c, p)\) during the design of software systems. Even though in [5] a model is defined which allows modeling of impreciseness using fuzzy sets, this is only for functional requirements. This means that at least six combinations of impreciseness are not addressed, with other types of impreciseness also possible.

To demonstrate how impreciseness can manifest itself and how it can be addressed, assume at a certain point a component needs to be designed. On this component certain quality requirements are imposed. For the remainder of this paper we will assume performance as the quality attribute. The requirements could then look like this

I. The component should on average react within \(a\) milliseconds

II. The component should react within a maximum of \(b\) milliseconds

We will define how to determine how well a certain alternative satisfies both requirements. We will use \(Q_1\) and \(Q_2\) to denote the amount to which the estimation satisfies requirement I and requirement II, respectively. Here \(Q_1\) and \(Q_2\) are functions of the type \((\text{Requirement}, \text{Estimation}) \rightarrow \text{Number}\). Both functions take a requirement and an estimation as input and return a value which represents the degree to which the estimation satisfies the requirement. Note that these individual quality attributes can be combined into one overall quality value by definition of a function that takes the priorities of the requirements in to account. Individual design alternatives can be compared based on these results.

The remainder of this paper is organized as follows: section 2 will describe how imprecise estimations for design alternatives can be evaluated with respect to crisp requirements, and section 3 will discuss this for imprecise requirement specifications. In section 4 the consistency of the approach is evaluated. Finally section 5 concludes the paper.

### 2. Evaluation functions for imprecise estimations with crisp requirements

Impreciseness in the estimations can manifest itself in different shapes and varieties. When these imprecise estimations are to be compared to the precise requirements, an approach is needed that can express the nature of the impreciseness in the estimations. For each type of impreciseness a definition should be made as to how they can be compared to the requirements. We have identified four possible types of impreciseness in the estimations: no impreciseness or crisp, probabilistic and fuzzy. Each of these types will be explained below, and also how they can be evaluated with respect to the requirements.

#### 2.1 Crisp estimations

In this most basic case all requirements are concise and all performance estimations are made using normal or crisp numbers. This could for instance mean that the system is expected to react within \(c\) milliseconds (where \(c\) is a number). This type of estimation can be directly compared to the requirements. Since the estimation either fulfills the requirement or not, the degree of fulfillment is either zero or one. The amount to which an individual requirement is fulfilled is defined as follows:

\[
Q_1(c, a) = \begin{cases} 
1, & \text{if } c \leq a \\
0, & \text{otherwise}
\end{cases} \quad Q_2(c, b) = \begin{cases} 
1, & \text{if } c \leq b \\
0, & \text{otherwise}
\end{cases}
\]

The values of \(Q_1\) and \(Q_2\) are determined by direct comparison of the estimated performance figure to the requirements. Whenever the estimation satisfies the requirement the respective quality attribute value is set to 1, otherwise it is set to 0.

#### 2.2 Probabilistic estimations

The performance is very difficult to estimate because of the load on a system. In this case we assume that the most accurate estimations can be made by using probabilistic
techniques. For each of the design alternatives a reasonably accurate performance estimate can be given based on the usage characteristics. The usage characteristics are supplied by performing a statistical market analysis of the current and future usage characteristics.

This type of analysis will most likely contain information on how many users will require the component to perform an operation at a given point in time. Probability distributions are used for modeling the uncertain nature of the arrival of requests. This means that the description will contain information on the distribution of requests and their respective probabilities. Given these distributions for each system a probability distribution or performance characteristic can be expressed.

It is not possible to directly compare distribution functions for system behavior to requirements that are expressed using crisp numbers. Since there are numerous possible response times for the system (each with their own respective probability of occurrence), it is not possible to determine with certainty whether the estimation satisfies the requirements. In other words we cannot give a yes (1) or no (0) answer. Rather we will use values between zero and one to describe which percentage of the return times is within the requirement restriction. Suppose the performance is estimated with a distribution function $f$. The average return time is $m$ given by:

$$m = \int_0^\infty xf(x)dx$$

The amount to which an individual requirement is fulfilled becomes:

$$Q_1(a, f) = \begin{cases} 1, & \text{if } m \leq a \\ 0, & \text{otherwise} \end{cases} \quad Q_2(b, f) = \frac{\int_0^b f(x)dx}{\int_0^\infty f(x)dx}$$

For the first requirement we can still determine whether the estimation is satisfactory, since the requirement specified an average response time. The average response time of a given distribution function can be determined and compared to the constraints given in the requirement in the same manner as the “crisp” case. For the second requirement the degree of fulfillment is calculated by determination of the fraction of the possible response times that is smaller than $b$.

2.3 Fuzzy estimations

An alternative form of impreciseness can manifest itself in performance estimations, which does not have a probabilistic character. The impreciseness for this type of estimation is a range of response times that are applicable, but some seem more applicable than others. This type of estimation can be expressed by using fuzzy sets. In a fuzzy set each element is member to a certain degree. The degree of membership is expressed using numbers between zero and one. With fuzzy sets it is for instance possible to describe fuzzy numbers. Suppose the performance is estimated with a fuzzy number with membership function $C$.

In figure 2 the fuzzy set $C$ is a triangular fuzzy number (the triangular shape is not obligatory, other types of fuzzy numbers can used if they are more applicable). In the picture also the restrictions $a$ and $b$ are indicated. Similar to the degree of fulfillment of requirements for the probabilistic estimations we can determine a degree of fulfillment for fuzzy estimations as well. This is defined in the following manner:

$$Q_1(a, C) = \frac{\int_0^a C(x)dx}{\int_0^\infty C(x)dx} \quad Q_2(b, C) = \frac{\int_0^b C(x)dx}{\int_0^\infty C(x)dx}$$

In this definition the degree of fulfillment is calculated by determination of the fraction of $C(x)$ that is smaller than or equal to $a$. The degree of fulfillment for the second requirement is calculated in a similar fashion. Note that both $Q_1$ and $Q_2$ can result in values between 0 and 1, which means that fuzzy estimations can model situations with more imprecision than the other alternatives.

2.4 Fuzzy probabilistic estimations

With the definition of imprecise requirements and estimations by use of probability distributions and fuzzy sets, the framework can address a wide range of specifications when comparing design alternatives. However, it is possible that a more refined method of expressing imprecision is needed. For instance, when a performance estimation should be made, it can be possible to determine the applicable distribution type, but not the exact parameters of the particular distribution. In this case the imprecision manifests itself in the accuracy of the specification of the distribution.

Recently there has been an increased interest in the fuzzy logic community in the area of fuzzy probabilities. In [3] fuzzy probability distributions are defined, by replacing...
parameters in families of crisp probability distributions (such as exponential, standard normal, etc.) with fuzzy values. This approach fits the refinement that is required for this particular type of impreciseness. Therefore a logical step is to include fuzzy probabilities in the framework to address this type of impreciseness.

For each type of probability distribution, such as exponential or standard normal, a set of parameters define the specific characteristics. Therefore a family of distribution functions of type \( f \) can be defined, which only vary in their parameters: \( \{ f_p | p \in P \} \), where \( P \subseteq \mathbb{R} \). If fuzzy sets are indicated with an overhead bar, a fuzzy probability distribution can be defined as follows: Suppose \( \bar{p} \) is a fuzzy set on \( P \). Then \( f_{\bar{p}} \) is a fuzzy probability distribution, from which the fuzzy expectation value \( G \) for a function \( g \) is obtained by:

\[
G(\alpha) = \left\{ \int_0^\infty g(x)f_{\bar{p}}(x)dx | p \in \bar{p}[\alpha] \right\}
\]

In this definition the \( \alpha \) in square brackets indicates an \( \alpha \)-cut. An \( \alpha \)-cut of fuzzy set \( A \) is given by [4]:

\[
A[\alpha] = \{ y | A(y) \geq \alpha \}
\]

Suppose the performance is estimated with a fuzzy distribution function \( f_{\bar{p}} \) for which the fuzzy expectation value for the average is given by:

\[
M(\alpha) = \left\{ \int_0^\infty xf_{\bar{p}}(x)dx | p \in \bar{p}[\alpha] \right\}
\]

Suppose in addition that \( DeFuz \) is a function that defuzzifies fuzzy sets to a number. The degree of fulfillment now becomes:

\[
Q_1(a, f_{\bar{p}}) = \frac{\int_0^a M(x)dx}{\int_0^a M(x)dx} \quad Q_2(b, f_{\bar{p}}) = \text{Defuz}(\bar{Q}_2)
\]

Where the fuzzy set \( \bar{Q}_2 \) is defined by:

\[
\bar{Q}_2[\alpha] = \left\{ \int_0^a f_{\bar{p}}(x)dx | p \in \bar{p}[\alpha] \right\}
\]

3. Evaluation functions for imprecise estimations with fuzzy requirements

Apart from impreciseness in the estimations, it is also possible for impreciseness to manifest itself in the requirements. The types of impreciseness differ somewhat from those of the estimations however. The nature of the requirement specification is either fuzzy or crisp. In the case a requirement specification contains probabilistic information this is also considered a crisp requirement, since there is no fuzzy information present. The parameters of the probability distribution are specified, which makes it possible to evaluate alternatives in a binary manner.

The potential fuzzy character of imprecise requirements is mostly expressed by use of terms like “The system should react in about 300 milliseconds” or “The average response time should approximately be 300 milliseconds”. Although the terms “about” and “approximately” contain information on the impreciseness, they do not have a numerical definition. To be able to evaluate estimations with respect to these imprecise requirements, the terms describing the impreciseness should be quantified. Since the values are not universal, but rather context dependent these fuzzy quantifications need to be determined based on domain knowledge. The representation of the imprecise information can be done in much the same manner as the fuzzy estimations by using fuzzy numbers.

The requirements should represent the inexact nature of the specifications given by the customer. For this purpose the restrictions \( a \) and \( b \) in the requirements I and II are fuzzy sets denoted \( A \) and \( B \) respectively. This fuzzy sets can have any form that seems to be applicable such as for instance triangular or trapezoidal fuzzy numbers.

3.1 Crisp estimations

In addition suppose for an alternative \( c \) a crisp number \( c \) is estimated as the performance characteristic.

In figure 3 the performance requirements \( a \) and \( b \) are depicted as triangular fuzzy numbers and the performance estimations for three alternatives are \( c, c' \) and \( c'' \) are indicated as lines. Logically the degree of fulfillment for requirement \( a \) should be \( Q_1(A, c) = 1 \), \( Q_1(A, c') = A(c') \) and \( Q_1(A, c'') = 0 \), and for \( b \) \( Q_2(B, c) = 1 \), \( Q_2(B, c') = 1 \) and \( Q_2(B, c'') = B(c'') \). However, it is not possible to
take the membership value of the estimation in A and B respectively. For for instance A(c) this would result in a value of 0, instead of the desired value of 1.

In addition to the fuzzy sets A and B fuzzy sets are needed that accommodate the values that are smaller than the smallest element still member of the fuzzy set. For each fuzzy set F there is a fuzzy set F’:

\[ F'(x) = \max \{ F(y) \mid y \geq x \} \]

This can be interpreted as: If F is a fuzzy number then F’ models the set of numbers which are smaller or equal than F. The amount to which an individual requirement is fulfilled by alternative c can now be defined as follows:

\[
Q_1(A, c) = A'(c) \\
Q_2(B, c) = B'(c)
\]

### 3.2 Probabilistic estimations

As has been described in section 2.2, it is also possible that estimations can only be made by using a probabilistic description. The probabilistic description can be compared to the fuzzy requirements in the following manner. Suppose a and b are fuzzy numbers with membership functions A and B, and suppose a distribution function f is estimated for the performance characteristic. The distribution is defined as follows:

\[
\int_0^\infty f(x)dx = 1 , \text{ with average } m \text{ given by: } \int_0^\infty xf(x)dx
\]

By using this definition the amount to which an individual fuzzy requirement is fulfilled is defined as follows:

\[
Q_1(A, f) = A'(m) \\
Q_2(B, f) = B'(m)
\]

For Q1 the degree of fulfillment is fairly straightforward since it is possible to determine the average response time from the probability distribution. Therefore we can take the membership value for the average of the distribution. For Q2 this is not possible, since a probability distribution does not specify the absolute upper limit of the possible reaction times. Instead longer reaction times will become increasingly more unlikely. Therefore the degree of satisfaction is defined as the expectation value of B’.

### 3.3 Fuzzy estimations

It is also possible that, like the requirements, the estimations also contain impreciseness in the form of fuzziness. By expressing the estimation using triangular numbers, similar to the requirements, deriving the degree of fulfillment is reduced to a comparison of fuzzy sets. In the literature many different methods for comparing fuzzy numbers have been proposed, each with their own merits and drawbacks. See [7] for a comparison of a multitude of such methods. To ensure transparency and consistency of the approach, a solution was selected based on the overlapping area of two triangular fuzzy numbers. Suppose a and b are fuzzy numbers A and B and suppose for an alternative a fuzzy number C is estimated.

![Graph showing fuzzy sets A, B, and C with overlapping areas for Q1 and Q2](image)

In figure 4 it can be seen how the degree of fulfillment can be calculated as the fraction of the area of C that overlaps with A’ or B’. The amount to which an individual requirement is fulfilled becomes:

\[
Q_1(A, C) = \frac{\int \min(A'(x), C(x))dx}{\int C(x)dx} \\
Q_2(B, C) = \frac{\int \min(B'(x), C(x))dx}{\int C(x)dx}
\]

For both Q1 and Q2 the degree of fulfillment is defined as the percentage of the area of the membership function of the estimation that coincides with the area of the respective requirement.

### 3.4 Fuzzy probabilistic estimations

As with comparing fuzzy probabilistic estimations to crisp requirements, the same can be done when comparing fuzzy probabilistic estimations to fuzzy requirements. Suppose the performance is estimated with a fuzzy distribution function \( f_p \) for which the fuzzy expectation value for the average is given by:

\[
M(x)[\alpha] = \{ \int_0^\infty xf_p(x)dx \mid p \in \tilde{P} \{ \alpha \} \}
\]
In addition suppose $a$ and $b$ are fuzzy numbers $A$ and $B$. The degree of fulfillment for the requirements is defined as follows:

$$Q_1(A, f_p) = \frac{\int_0^\infty \min(A'(x), M(x))dx}{\int_0^\infty M(x)dx}$$

$$Q_2(B, f_p) = \text{Defuz}(\overline{Q_2}(B, f_p))$$

where $\overline{Q_2}(B, f_p)[\alpha] = \left\{ \int_0^\infty B'(x)f_p(x)dx \mid p \in \overline{P}(\alpha) \right\}$

4. Consistency Analysis of the Approach

The approach presented in this paper identifies two types of imprecision in the requirements and four types of imprecision in the estimations. This leads to a total of eight possible combinations, for each of which a method for evaluation has been defined. However, for approach to be consistent in specific inputs the comparisons should behave exactly like their “simpler” counterparts. For instance, when a distribution is defined as a probability of occurrence of zero for all reaction times except one, which has a probability of one, this is essentially a crisp input. Therefore the result after evaluating this distribution should be equal to the evaluation of the crisp case. The choice for each of the individual cases can be done as follows:

**Probabilistic to Non-Probabilistic**

When choosing a probability distribution that behaves exactly as a non-probabilistic case, the approach should behave as with a non-probabilistic problem. In this case the $\delta$-distribution can be chosen, which is defined by:

$$\int_0^\infty g(x)\delta(x-a) = g(a)$$

for any given function $g$. The $\delta$-distribution is a generalized distribution, which peaks at 0.

**Fuzzy to Crisp**

Similar to the probabilistic case a fuzzy set can be constructed that contains only one element, and this element has a membership value of one, commonly termed a crisp singleton set.

**Fuzzy Probability to Crisp Probability**

Finally the fuzzy probabilistic case will revert to a normal probabilistic case, since the fuzziness here describes impreciseness with respect to which distribution is applicable. Once again this can be done by defining a crisp singleton set, which contains only one set of distribution parameters with membership value one.

By choosing these inputs the results become identical to the “simpler” counterparts, non-probabilistic, crisp or probabilistic respectively. Due to lack of space the complete proofs for each individual case will not be given, but are left to the reader.

5. Conclusion

In this paper an approach has been presented which enables the software engineer to express estimations in an accurate manner. By applying techniques from probability theory and fuzzy logic the nature of the impreciseness in the estimation can be described. The impreciseness might also manifest itself in the requirements themselves.

In addition to means for describing impreciseness in both requirements and estimations, the approach also defines mechanisms for comparing imprecise specifications. These mechanisms enable the software engineer to evaluate design alternatives with respect to the requirements, even if one or both incorporate impreciseness.

6. References


