VILNIUS UNIVERSITY

JOLANTA MILIAUSKAITĖ

A FUZZY INFERENCE-BASED APPROACH TO PLANNING QUALITY OF ENTERPRISE BUSINESS SERVICES

Doctoral Dissertation

Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2015

The dissertation was written between 2010 and 2014 at Vilnius University Institute of Mathematics and Informatics.

Scientific Supervisor:

Prof. Dr. Albertas Čaplinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

VILNIAUS UNIVERSITETAS

JOLANTA MILIAUSKAITĖ

NERAIŠKIAIS SAMPROTAVIMAIS GRINDŽIAMAS METODAS VERSLO PASLAUGŲ KOKYBEI PLANUOTI ĮMONIŲ PASLAUGŲ STILIAUS INFORMACINĖSE SISTEMOSE

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07 T)

Vilnius, 2015

Disertacija rengta 2010-2015 metais Vilniaus universitete Matematikos ir informatikos institute.

Mokslinis vadovas:
prof. dr. Albertas Čaplinskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor Prof. Dr. Albertas Čaplinskas for sincere knowledge sharing, collaboration, consultancy and indispensable support and guidance throughout the process of this dissertation in this exciting field. He managed to find the time and energy to read probably every draft of this dissertation and untangle “crosscutting concerns” of my scientific writing. I am very grateful to him for pushing further my ideas and significantly increasing the quality of my work.

I am also grateful to the stuff of Vilnius University Institute of Mathematics and Informatics, especially Prof. Dr. Gintautas Dzemyda, Prof. Dr. Saulius Gudas, Assoc. Prof. Dr. Audronė Lupeikienė, and Laima Paliulionienė for the patience, constructive feedback, sincere interest, and valuable advices. I am also sincerely thankful to Assoc. Prof. Dr. Vytautas Čyras from Vilnius University Faculty of Mathematics and Informatics, for his constructive feedback.

I wish to thank to my family, my husband Andreas, my children Julius Vytautas and Joris Vasaris, for their support and endless patience.

I would like to express my thanks to all the people who have been in one way or another involved in the preparation of this dissertation, for all of their support and tolerance during this challenging period of my life.

Abstract

The concept of service-oriented architecture (SOA) becomes increasingly popular in the field of information systems as well as software systems engineering. The symbiosis between an Enterprise Architecture and SOA results in so-called Service-Oriented Enterprise Architecture. Systems that implement this architecture are addressed as service-oriented enterprise systems (SoES). They are composed of enterprise business services (EBS), i.e. components implementing some business logic that is embedded in web services. EBS is not a software product because it is a service. This raises a number of new software engineering problems including problem of assessment of investments necessary for the development and implementation of new EBSs. In order to assess required investments, the quality of each EBS (QoS_{EBS}) should be planned at least roughly. It is far not simple problem because a number of stakeholders, which interpret the concept of QoS_{EBS} differently and plan it from different role-dependent perspectives, are involved in this process. In addition, a right conceptual basis for describing QoS_{EBS} planning algorithms and even QoS_{EBS} itself still is absent.

The present dissertation contributes to the solution of this problem in a following way:

- a conceptual framework (or a sort of ontology), which forms a conceptual basis for our theoretical research and enables to describe SoES, EBS, QoS_{EBS}, QoS_{EBS} planning problem and models, methods and algorithms for solving this problem in a precise consistent way, is developed;
- QoS_{EBS} planning problem as a mathematical problem is described and a methodology to guide this problem fuzzification process is proposed;
- a problem-oriented QoS_{EBS} model is build and an ensemble of collaborating methods inspired by this model and expressed in a form of detailed computational algorithms (further, an ensemble of collaborating algorithms) are designed;

- a software architecture that implements the ensemble of collaborating algorithms to solve the QoS_{EBS} planning problem is developed and described by a set of UML diagrams.

To evaluate the research results, an exploratory implementation of QoS_{EBS} planning system was developed, a case study controlled experiment was performed, and results of this experiment were validated w.r.t. internal, external and construct validities. The experiment proved acceptability of the proposed approach as a whole and computational correctness of the ensemble of QoS_{EBS} planning methods and algorithms.

Contents

List of Figures

List of Tables

Content of CD Attached to the Dissertation

Appendix B. QoS_{EBS} Planning System Program Text

Appendix C. API Technical Documentation

Appendix D. Results of Case Study 1: Fuzzy Reasoning

Appendix E. Results of Case Study 2: Linguistic Approximation

Appendix F. Results of Case Study 3: Perspective Aggregation

Appendix G. Results of Case Study 4: Viewpoint Aggregation

Glossary

Chapter 1 Introduction

1.1. Research Context

The symbiosis between Enterprise Architecture (EA) and Service-Oriented Architecture (SOA) results in the so-called Service-Oriented Enterprise Architecture (SoEA) and brings up new problems for Service-Oriented Enterprise Systems (SoES) engineering. In the context of SoES, a number of different stakeholders exist. They have different, often conflicting understandings of a SoES business service quality and different opinions on how such a service should be defined, specified, and evaluated. Thus, despite a large number of existing QoS models and ontologies, the question "What does QoS mean?" still has no final answer, at least, in the context of SoES. This concept is still not well-defined even for the traditional SOA service.

Originally, the term QoS was introduced in telecommunication where it was focused on the service performance measures from the network perspective. Later on, it was extended including even hardly related to quality characteristics such as a service requestor's satisfaction or service cost.1 Currently the term QoS refers to several different things. As stated in Benbernou et al. [1]

> *"This set of quality attributes does not characterize only the service but any entity used in the path between the service and its client. Such an entity may exist in any of the three possible service levels. Thus, different QoS attributes may be used to define the QoS of a service in the application, service, and infrastructure levels".*

In this quote, the term client refers to the service requestor. In the text of this dissertation, instead of the term "service level" we use the term "perspective" (see Definition 54 in Chapter 4).

[1] 1Note, that in this work, the terms *service requestor* and *service consumer* mean different things. The first one is a person or an organisation while the second one is a piece of software. In a similar way, the term *service owner* refers to a person or organization, whereas the term *service provider* refers to a piece of software.

Thus, to answer the question, what is the meaning of the term "*Quality of Service*", is far from being a simple task. This concept still remains murky. Its definition strongly depends on the topic of discourse and is usually not adaptable to other discourses. Moreover, even the quality itself is a vague and highly subjective concept [2] and, to date, no commonly agreed understanding of this concept has existed.

This situation raises a number of research questions related to the nature of QoS as well as to the planning, prediction, and evaluation of a particular SOA service. From the practical point of view, it is especially important to answer these questions in the context of SoES.

The first thing that should be done to achieve this aim is to investigate the nature of the quality of enterprise business service (QoS_{EBS}) and to define this concept more precisely. In turn, to do that, it is necessary to make analysis of different understandings of QoS and QoS_{EBS}, to systematise and to generalise these understandings and to relate them each to other. It means that a theory of meaning should be developed for the QoS_{EBS}. As pointed out by James McGilvray in [3], a theory of meaning is understood by most as a theory that focuses on word-world relationship whether it is referential or alethic. The term alethic (Greek *aletheia* 'truth') is defined as follows: "Of or pertaining to the various modalities of truth, such as the possibility or impossibility of something being true."2 So an alethic theory of meaning [4] is a theory which deals with the truth and correctness of sentences or perhaps propositions. In the referential theory of meaning, a term denotes objects, facts are relations between objects, expressions capture the relations of objects, and a true statement corresponds to facts.

Despite the fact that the referential theory of the meaning of QoS_{EBS} in many aspects is considerably simpler than the natural language meaning theory, sometimes it is still rather complicated to define a regular word-world relationship because it is context-dependent and any concept "can be used by people in whatever context they happen to be in to serve any number of

[2] Philosophy Dictionaryhttp://philosophy.enacademic.com/79/alethic

purposes and to refer to any number of things” [3]. For example, the same QoS_{EBS} property, reliability can mean different things for different EBS.

In the alethic meaning theory, constraints impose necessities that cannot, even in principle, be violated typically because of some physical or logical law. Alethic propositions are interpreted in terms of Kripke’s possible world semantics. In the SoES context, a proposition is necessarily true, if and only if it is true in all possible worlds. With respect to EBS static constraint, declared by the service level agreement (SLA) for a given EBS, a possible world corresponds to the state of the fact model that might exist at some point of time. A proposition is possible, if and only if it is true in at least one possible world. Impossible propositions are true in none of the possible worlds (i.e. false in all possible worlds). However, problems with the alethic theories of meaning also arise because the truth values of the same QoS_{EBS} property may differ for different EBS.

The theory of QoS_{EBS} meaning is a conceptual basis for an empiric QoS_{EBS} planning, prediction, and evaluation theory that should provide models, methods, and algorithms to solve the related planning, prediction and evaluation problems. The dissertation seeks to contribute to this field by developing a fragment of such an empiric theory aimed at the preliminary planning of QoS_{EBS} for particular EBS and a fragment of the QoS_{EBS} meaning theory, required to describe the related models, methods, and algorithms. The research is rather exploratory research and does not pretend to an exhaustive investigation of all the mentioned issues in detail.

1.2. Problem Statement

In an informal way, the problem investigated in the dissertation is formulated as follows (Fig. 1):

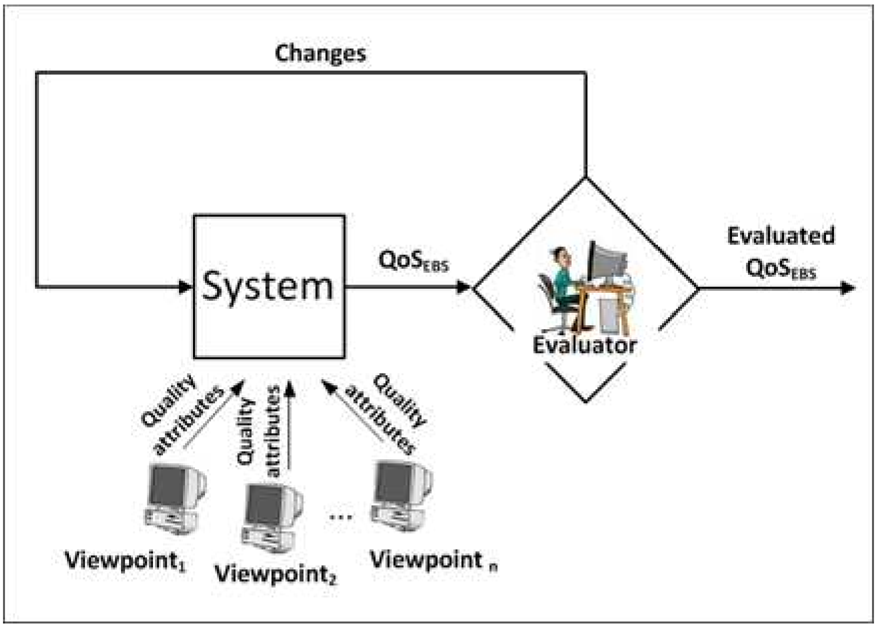


Fig. 1 The intuitive formulation of the investigated problem

"*Giving role-depended, subjective and conflicting preliminary requirements, that define an acceptable quality level of bottom level quality sub characteristics for planned to be implemented in a particular web-based SoES, an enterprise business service, specified by a number of different stakeholders, to resolve conflicts and to define preliminary requirements for the acceptable quality level of EBS quality characteristics*".

The formal definition of this problem is presented in the section 6.1.

1.3. Motivation

When developing a new EBS, its QoS_{EBS} should be planned, at least roughly, taking into account the conflicting quality requirements, stated by different stakeholders, as well as constraints of financial, time and other resources. Conflicts between the requirements arise not only due to the role-dependent perspectives (service owner, computer network administrator, infrastructure administrator, service user, etc.), from which different stakeholders see this new service, but also due to a different understanding of the quality concept itself or, in other words, due to different viewpoints to the quality. Although preliminary quality requirements are necessary only to approximately estimate the investments, required for the development of this EBS, and can be defined only roughly, the conflicts should be resolved in some way and a solution acceptable to all parties should be found. Usually it is done during a long-lasting and time-consuming negotiation process. So, it is very desirable to replace this process by an appropriate software system. However, as mentioned in Section 1.1, the existing theoretical basis is not mature enough for the development of such software and, consequently, some theoretical research should be performed for its improvement.

1.4. Aims and Objectives of the Research

The overall aim of this research is to develop a fuzzy inference-based QoS_{EBS} planning approach and supporting software system, such that, taking into account financial, time, and other resource constraints of a project, we could solve the problem stated in Section 1.2, and thus help the enterprise to roughly plan the investments required to develop planned EBS. In order to achieve this aim, the following research objectives were set up:

1. to develop a system of related concepts (a conceptual framework), which provides a conceptual basis for a theoretical part of our research enabling us to describe in a precise and consistent way the SoES, EBS, QoS_{EBS}, the QoS_{EBS} planning problem, and models, methods and algorithms for solving this problem;
2. to formalise the QoS_{EBS} planning problem, stated in Section 1.2, and to propose a methodology to guide the fuzzification process of this problem;
3. to build a QoS_{EBS} model suited to solve the QoS_{EBS} planning problem and to design an ensemble of collaborative methods, inspired by this model and expressed in the form of detailed computational algorithms;
4. to develop software architecture that implements an ensemble of collaborative algorithms for solving the QoS_{EBS} planning problem.

1.5. Assumptions

The applicability of research findings is limited by the following a priori stated assumptions:

1. Fuzzy set theory and fuzzy logic methods are most appropriate for a preliminarily planning enterprise business service of the acceptable quality level.

   Motivation: The meaning of EBS qualities cannot be defined by formally defined properties because they are vague concepts. For example, the meaning of quality '*reliable*' depends on a particular interpretation of the concept *quality* and the meaning of the acceptable quality level "high quality" cannot be defined precisely at all because the meaning of the word '*high*' depends on a particular context.

2. Enterprise business service is intended to be used in the SoES environment.

   Motivation: SoES operates in an enterprise controlled intranet/extranet environment. SOA systems operate in an open internet-wide environment, in which infrastructure, networks, and other resources cannot be controlled by an enterprise and, consequently, QoS is of probabilistic nature. In an open environment, the proposed planning approach is not applicable.

3. The conflicts among the quality requirements, stated by stakeholders, should be resolved taking into account the importance of each role.

   Motivation: Stakeholders are not peers. They make a different impact on the decision about the acceptable QoS_{EBS} quality level. The importance of a service owner, service requestor, network administrator and other stakeholders depends on both a particular SoES and particular EBS. So, conflicts among EBS quality requirements should be resolved taking into account the weight of each stakeholder.

4. The conflict resolution procedure defines the resulting quality as a fuzzy set X. This result must be interpreted in terms of linguistic labels that describe possibly acceptable quality levels (*high quality*, *moderate quality*, etc.) for considered EBS. In other words, the linguistic

approximation of X should be performed. The linguistic approximation procedure approximates X by a fuzzy set Y_i, representing some linguistic term T_i of a linguistic variable L (further denoted by $L.T_i$). To this end, it uses a similarity measure $m(X,Y_i)$. It may happen that for several linguistic terms this measure has the same value. In such cases, the decision should be made by a decision-maker using an interactive procedure.

Motivation: The main recommendation [5] is to refine the similarity measure and make it more sensitive. However, in the context of our research, where the number of linguistic terms is not high and linguistic modifiers are not used, an interactive procedure is more practical.

1.6. Research Questions

The dissertation aims to answer the following research questions:

1. How is QoS_{EBS}, defined taking into account different understandings of the concept “quality”?
2. Which one of the current approaches to service quality modelling (if any) is likely to be suitable for modelling QoS_{EBS}?
3. What strategy should be used to construct the most suitable membership functions (MF) in order to fuzzify the concept of QoS_{EBS} for particular EBS?
4. What kind of fuzzy reasoning formalism is best suited for inferences in the tree structures, that describe hierarchy of QoS_{EBS} properties, and which algorithms are most suitable to implement such inferences?
5. Which methods and algorithms should be used to resolve conflicts among QoS_{EBS} requirements, stated by different stakeholders?
6. What kind of ensemble of collaborative algorithms is most suitable to solve the QoS_{EBS} planning problem?
7. What type of architecture is needed for the software system that implements this ensemble of collaborative algorithms?

1.7. Research Hypotheses

The following hypotheses have been justified in the dissertation too:

H1. Despite the differences between the nature of services and products, all currently identified understandings of a product quality are also applicable to a service.

H2. A fuzzy inference-based approach to the planning quality of enterprise business services enables us to define QoS_{EBS}, taking into account different understandings of the concept “quality”.

H3. Not a single of the current approaches to service quality modelling is suitable to model QoS_{EBS} in a suited to design effective ensemble of collaborative methods to resolve conflicts among requirements, stated by different stakeholders.

H4. Such a QoS_{EBS} model can be develop only by using a fuzzy inference-based approach to the planning quality of enterprise business services.

H5. The fuzzification of a particular QoS_{EBS} property depends on both the nature of this property and on the EBS consideration perspective. However, it is possible to develop a methodology that guides the fuzzification procedure for any QoS_{EBS} property and any EBS consideration perspective.

H6. In the context of QoS_{EBS} planning problem, a fuzzy reasoning formalism, that combines semantic derivation and aggregation techniques, is acceptable for inferences in tree structures, which describe the hierarchy of QoS_{EBS} properties, because it meets all the functional requirements.

H7. In order to solve the QoS_{EBS} planning problem, it is suffices that an ensemble of collaborative algorithms combines problem fuzzification, balancing, fuzzy reasoning, linguistic approximation, and fuzzy aggregation algorithms.

H8. Solving the QoS_{EBS} planning problem, the quality of results, delivered by EBS (i.e. a product presented for the service requestor), should also be taken into account.

H9. Object-oriented software system architecture is sufficient to implement and ensemble the collaborative algorithms, which resolve conflicts among quality requirements, stated by different stakeholders, and solves the QoS_{EBS} planning problem.

According to Merriam-Webster Dictionary[3], the verb '*justify*' means "*to prove to be just, right, or reasonable*". According to [6], the act of '*hypothesis justification*' is an attempt to convey and convince oneself and/or others about the validity of the stated hypothesis or, in other words, to demonstrate by some evidence that the stated hypothesis is correct, defensible, and, probably, acceptable. A justified hypothesis is referred to as a confirmed hypothesis. A hypothesis can be justified in a number of different ways: proving by construction procedure [7] (applicable to H2, H4, H5, H8, H9), by an evidence-based inductive reasoning procedure [8] (applicable to H1, H3, H6), by a case-based controlled experiment [9] (applicable to H7), by a statistical testing procedure [10], disproving by a counterexample [7] (applicable to H1, H3, H6) or using some hypothesis validation and verification procedures.

1.8. Research Design and Research Methods

1.8.1. Overall Research Design

According to [11], a research design "serves as the architectural blueprint of a research project". In other words, it refers to the plan, structure, and strategy of research, and guides the whole research process in so that to obtain answers to research questions. In short, it is the overall scheme of the research, which outlines what the investigator should do and how he/she will do that. It does not describe the research in detail, only suggests the main course of actions. The type of research design depends on the kind of research to be done. In the dissertation, an exploratory research is carried out. The research focuses on the question "How?" or, to be more exact, on the question "How to plan QoS_{EBS} taking into account role-dependent, subjective and conflicting preliminary requirements stated by different stakeholders?" Exploratory research is

[3] http://www.merriam-webster.com/

research that addresses a subject with a high level of uncertainty. It investigates the problem that is not very clear because very little research on this subject has been carried out. According to [12],

> *"Exploratory research contributes to the continued vitality of every discipline. The aim of exploratory research is to identify new tasks-tasks that cannot be solved by existing methods. Once a new task has been found, exploratory research seeks to develop a precise definition of the task and to understand the factors that make the task different from previously-solved tasks."*

To date, the most part of PhD research in information systems and software engineering are still of exploratory nature. The research performed in the dissertation preliminarily was inspired by: a) the previous research on the quality of service [13,14,15] pursued in the Software Engineering Department of the Institute of Mathematics and Informatics (currently, Vilnius University Institute of Mathematics and Informatics); b) the perspective integration techniques proposed in the traditional requirement engineering [16,17,18,19] and business process modelling [20], enterprise modelling [21], architecture engineering [22], development of distributed applications [23]; and c) some ideas of i* methodology [24,25,26]. On this basis, a preliminary problem statement, research assumptions and research hypothesis were formulated (Step 1 in Fig. 2) as well as a research design has been developed. During the investigation, the problem statement, assumptions and hypothesis were redefined many times, the research strategy was modified, and the research was redesigned. In other words, the investigation was performed in the incremental manner. It is typical of any exploratory research because it is incremental by definition. The research strategy, defined by the final research design, is presented in Fig. 2. Let us briefly consider the main steps of the strategy.

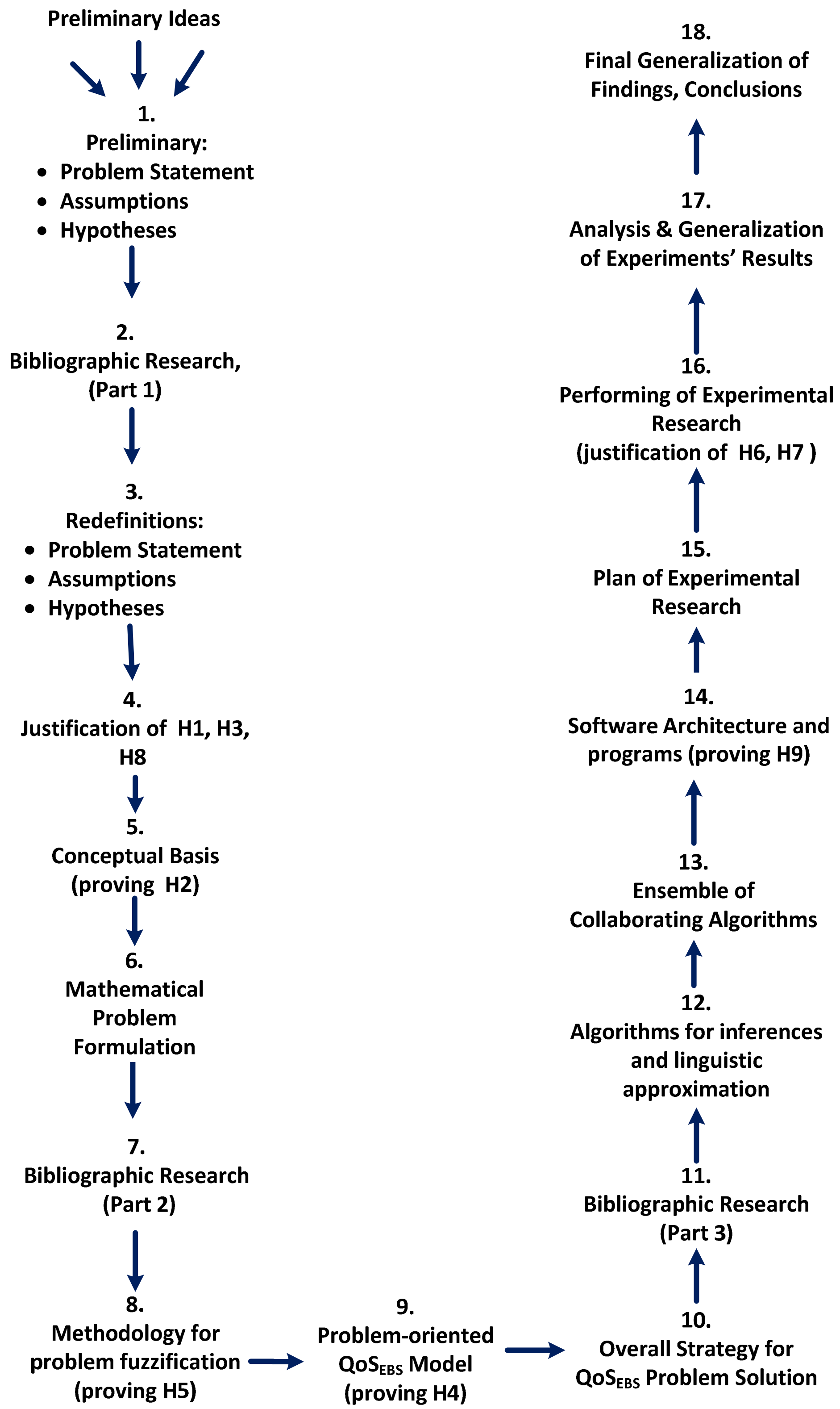


Fig. 2 The research strategy defined by the final variant of research design

In Step 2, the first part of the bibliographic research [27] was performed. The end of this research is to survey the literature on the quality, first at all, service quality issues. The obtained results are used for two aims:

- Step3: to refine and redefine the problem statement, research assumptions and hypotheses.
- Step 4: to justify hypotheses H1, H3, and H8.

In Step 5, the conceptual framework, which forms a conceptual basis of the research, was developed and hypothesis H2 was proved thereby by construction. In Step 6, the QoS_{EBS} planning problem was reformulated as a mathematical formulation problem. In Step 7, the second part of the bibliographic research was fulfilled. The aim of this research was to make a conceptual analysis of literature sources on the membership function construction and to collect materials, necessary for developing the methodology to guide the QoS_{EBS} planning problem fuzzification process. The methodology itself was developed in Step 8 and hypothesis H5 was proved thereby by construction. In Step 9, a QoS_{EBS} model, suited to solve the QoS_{EBS} planning problem, was developed and hypothesis H4 was proved thereby by construction. In Step 10, the overall strategy was developed to solve the QoS_{EBS} planning problem. In Step 11, the third part of the bibliographic research was performed. The aim of the research was to make a conceptual analysis of fuzzy reasoning formalisms and to choose such that is best suited for inferences in tree structures that describe the hierarchy of QoS_{EBS} properties. In Step 12, an algorithm for inferences in the mentioned tree structures was developed. In step 13, an ensemble of collaborative algorithms, required to solve the QoS_{EBS} planning problem, was designed. In Step 14, software architecture that implements this ensemble and appropriate software programs, were developed and hypothesis H9 was proved thereby by construction. By an ensemble of collaborative algorithms we mean a problem-oriented arrangement of a collection of algorithms which: a) implements a given collaboration pattern; b) is designed taking into account the computational resource constraints and the specific features of a class of problems that should be solved; c) optimizes the overall performance of a system implementing this ensemble; and d) seeks to find the required solution in a reasonable time. In Step 15, a plan of experimental research and a detailed methodology of the research were developed. In Step 16, experimental research was performed. The aim of the research was to justify hypotheses H6 and H7. In Step 17, the results of the performed experiments were analysed and generalized.

Finally, in Step 18, the final generalization of the dissertation research findings was performed, conclusions were drawn, and the text of the dissertation and its summary was prepared for publishing.

1.8.2. Research Methods

A mixed methods approach that combines quantitative and qualitative research methods was chosen for the research. According to [28],

> "*…a mixed methods approach is one in which the researcher tends to base knowledge claims on pragmatic grounds (e.g., consequence-oriented, problem-centred, and pluralistic). It employs strategies of inquiry that involve collecting data either simultaneously or sequentially to best understand research problems.*"

In the mixed methods approach, the data collection involves gathering both numerical data (e.g., experimental) as well as text information (e.g., collected from literature) so that both quantitative and qualitative information is used in the problem solving procedures.

The bibliographic research methodology was intensively used in Steps 2, 7 and 11. However, different methods of the bibliographic research were used in different steps. In Step 2, the Systematic Literature Review (SLR) method [29,30,31,32,33] and interpretative data synthesis [34,35] of the collected qualitative data were used for this aim.

SLR is "*an efficient scientific technique to identify and summarise evidence on the effectiveness of interventions and to allow the generalizability and consistency of research findings to be assessed and data inconsistencies to be explored*" [36]. This method focuses on the stated research question and helps to identify, select, and critically evaluate all high quality research evidence relevant to this question. For each research question, it is necessary to prepare a review protocol, which sets out a detailed review plan and defines search criteria, inclusion and exclusion criteria, the selection process, quality assessment, the data extraction process, data synthesis, etc. Review protocols should be formed after specifying the research questions. The research questions stated for the part 1 of the performed bibliographic research, review protocol and search results are presented in Appendix A.

Qualitative data collected as a result of SLR (texts, in our case) must be combined, generalised and interpreted, or, in terms of SLR methodology, synthesised [34]. It is not a simple task because different authors belong to different theoretical, methodological and terminological traditions, their research is based on different ontological assumptions, guided by different philosophical assumptions and sometimes their claims can be interpreted in several different ways. On the other hand, the combination of findings of different scientific schools is the strength of any literature review.

Generally, there are two approaches to data synthesis: *integrative* and *interpretative* [37]. The integrative reviews are primarily suitable for synthesising quantitative data, but, in cases where the concepts, under which the findings are to be summarised, are well defined and explored [38]. However, this approach is not applicable in the case of an exploratory research, where the concepts are ill-defined or not defined at all. On the other hand, the interpretive synthesis deals with the development of concepts, as well as with the development and specification of theories that integrate those concepts. According to [38],

> *"An interpretive synthesis will therefore avoid specifying concepts in advance of the synthesis. In contrast with an integrative synthesis, it will not be concerned to fix the meaning of those concepts at an early stage in order to facilitate the summary of empirical data relating to those concepts. The interpretive analysis that yields the synthesis is conceptual in process and output, and the main product is not aggregations of data, but theory."*

For this reason, we have chosen the interpretative synthesis and, to be more exact, applied three techniques of this method: *reciprocal translation analysis*, *descriptive synthesis*, and *refutational synthesis*. Reciprocal translational analysis is based on the “translation” of concepts from individual studies into one another, evolving thereby overarching concepts [37,34]. The descriptive synthesis is used to subsume a number of complimentary concepts [35]. Finally, the refutational synthesis is used to explore and explain contradictions between individual studies [37,39].

For the bibliographic research in steps 7 and 11, the focused literature review [40,41] was used. In Step 7, the focus was on problem fuzzification methods (mainly, on membership function construction approaches) and in Step 11, on different fuzzy reasoning formalisms and approaches. In both cases, the literature review was combined with the constructive research [42,43] performed in steps 8 and 12, respectively. Constructive research is a research procedure for producing innovative constructions, intended to solve the problems encountered in the real world, and to make some contribution to the theory of the discipline in which it is applied ([42,43]. The central notion of this approach, the novel construction, is an abstract notion with a great variety of potential realizations. Models, designs, methodologies, algorithms, and many other artefacts are considered as constructions. It means that they are invented and developed, not discovered. In Step 8, the constructive research was used to prove hypothesis H5 by construction. In line with the constructive research approach, the proof by construction [44] is the one in which an object that proves the truth value of a statement is built, or found. There are two main uses of this technique: a) proof that a statement with an existential quantifier is true; and b) disproof by counterexample: this is a proof that a statement with a universal quantifier is false. In Step 8 technique a) was used. In Step 12, the aim of constructive research was to choose suitable fuzzy reasoning formalism, and best fuzzy inference and linguistic approximation algorithms. For this aim, the technique b) was used. In other words, a number of fuzzy inference and linguistic approximation algorithms were considered and tested by the construction of counterexamples.

The constructive research methodology was also used in Step 9 to build and to investigate the problem-oriented QoS_{ESB} model and prove hypothesis H4, thereby.

For the development of a conceptual framework in Step 5, the theory building methodology [45] was used. The most important theoretical result of our research is a fragment of an exploratory theory [46] for planning of QoS_{EBS}. Theory building is treated as an attempt "to incorporating all that is known from the current literature (theoretical, mathematical, empirical, and practitioner research) into a single, integrated consistent body of knowledge" [47]. Any theory includes a component referred as its conceptual framework. According to [48], a conceptual framework can be defined as follows:

> "*…a network of interlinked concepts that together provides a comprehensive understanding of a phenomenon or phenomena. The concepts that constitute a conceptual framework support one another, articulate their respective phenomena, and establish a framework-specific philosophy. Conceptual frameworks possess ontological, epistemological, and methodological assumptions, and each concept within a conceptual framework plays an ontological or epistemological role. … The methodological assumptions relate to the process of building the conceptual framework and assessing what it can tell us about the "real" world.*"

A number of approaches and methods exist for building of conceptual frameworks. We use for that conceptual methods [49], including conceptual analysis [50]. According to [50], a conceptual analysis is the analysis of concepts, terms, variables, constructs, definitions, assertions, hypotheses, and theories. It involves examining these for clarity and coherence, critically scrutinizing their logical relations, and identifying assumptions and implications. The goal of conceptual analysis is to increase the conceptual clarity of the research subject.

Finally, a case-based controlled experiment methodology [51,52] was used in the experimental part of the dissertation research (steps 15-17). The experimental research is described in detail in Section 7.2.

1.8.3. Instrumentation

The term *instrumentation* refers to the software and other tools used in the data-collection process, experimental research and other research activities. It is related not only to instrument design, selection, construction, and assessment, but also to the conditions under which the instruments are used. Not proper research instruments or not proper usage of these instruments may lead to biased results. Therefore, instrumentation is also a specific term with respect to a threat to internal validity of research. [53].

In the dissertation research, the instrumentation was used in two steps of research design, namely, in Step 2 (Part 1 of bibliographic research) and Step 16 (experimental research). The instruments, used in the bibliographic research, and conditions under which they were used, are described in detail in Appendix A. Bibliographic Research. The instrumentation of the experimental research is described in detail in Section 7.1.

1.9. Results

The results of the dissertation can be summarized as follows:

[1] Conceptual framework, that forms a conceptual basis for further theoretical research and enables us to define, in a precise and consistent way, SoES, EBS, QoS_{EBS}, QoS_{EBS} planning problems and models, methods and algorithms for solving this problem;

[2] Formulation of the QoS_{EBS} planning problem as a mathematical problem and a methodology to guide the problem fuzzification process;

[3] The problem-oriented QoS_{EBS} model and an ensemble of collaborating methods inspired by this model and expressed in a form of detailed computational algorithms;

[4] Software architecture that implements the ensemble collaborating algorithms to solve the QoS_{EBS} planning problem (described by a set of UML diagrams).

1.10.Scientific Contribution of the Research

The research conducted is one of the first that investigates how to apply the fuzzy set theory and fuzzy logic in the formalisation and planning of QoS_{EBS} (partly, also of QoS_{SOA}). Its scientific contribution to the field of informatics engineering is threefold: a) a conceptual framework enabling us to describe the SoES, EBS, QoS_{EBS}, the QoS_{EBS} planning problem in a precise and consistent way, and models, methods and algorithms for solving this problem have been developed; b) the research shows that well-known fuzzy logic methods (e.g. Mamdani implication), used to implement fuzzy controllers and to solve various diagnostic problems, are not suitable for quality planning problems; c) the research has demonstrate how to combine semantic derivation and aggregation methods in the implementation of linguistic fuzzy reasoning procedures.

1.11.Practical Value of the Research Results

The problem-independent methodology to guide the problem fuzzification process, developed in the dissertation research, can be applied in many industrial projects. The software architecture, that implements an ensemble of collaborative algorithms for solving the QoS_{EBS} planning problem, can be used as a kind of reference architecture in projects, aimed at implementing ensembles of collaborative algorithms.

1.12.Approbation

The main results of the dissertation were presented, discussed and approved at the following international and local conferences, doctoral consortiums, and workshops:

- The 11th International Baltic Conference on Database and Information Systems [Baltic DB&IS 2014], June 8-11, Tallinn, Estonia;

- The 2nd International Business and Systems Conference BSC 2013, Riga Technical University, November 6–7, 2013, Riga, Latvia;
- The Tenth International Baltic Conference on Databases and Information Systems (Baltic DB&IS'2012), Doctoral Consortium, July 8–11, 2012, Vilnius, Lithuania;
- The 2nd International Doctoral Consortium Informatics and Informatics Engineering Education Research: Methodologies, Methods, and Practice, November 30–December 4, 2011, Druskininkai, Lithuania;
- The 55th Conference of the Lithuanian Mathematicians Society, June 26–27, 2014, Vilnius, Lithuania;
- The 6th International Workshop of Data Analysis Methods for SoftwareSsystems [DatAMSS], December 4–6, 2014, Druskininkai, Lithuania;
- The 5th International Workshop Data Analysis Methods for Software Systems, December 5–7, 2013, Druskininkai, Lithuania;
- The 15th Conference of the Lithuanian Computer Society “Computer Days–2011”, September 22–24, 2011, Klaipėda, Lithuania.
- The 16th Conference of the Lithuanian Computer Society “Computer Days–2011”, September 19–21, 2013, Šiauliai, Lithuania.

1.13.Publications

The main results of the dissertation research were published in the following scientific publications.

Journal publications

1. Lupeikienė, A., Miliauskaitė, J., Čaplinskas, A. (2013). A Model of View-Based Enterprise Business Service Quality Evaluation Framework. *Informatica*, Vol. 24, Iss. 4, p. 543-560.

   Research Contribution: In review of related works and problem fuzzification approach, Paper writing contribution: medium

2. Miliauskaitė, J. (2015). Some Methodological Issues Related to Preliminary QoS Planning in Enterprise Systems. Baltic Journal of Modern Computing, Vol. 3, No. 3, pp. 149-163.

   Research Contribution: High, Paper writing contribution: high (main author)

3. Miliauskaitė, J., Čaplinskas, A. (2011). Modularization of Web Service Composition. *Information sciences*, Vol. 56, p. 73-84 (in Lithuanian).

   Research Contribution: high, Paper writing contribution: high

4. Lupeikienė, A., Miliauskaitė, J., Čaplinskas, A. (2013). Critical analysis and elaboration of three prevailing approaches to model quality of service. *Information sciences*, Vol. 65, p. 111-119.

   Research Contribution: medium, Paper writing contribution: high

Publications in Proceedings

5. Miliauskaite, J. (2014). The Membership Function Construction in View-based Framework. In *Proceedings of the 11th International Baltic Conference on Database and Information Systems* [Baltic DB&IS 2014], Tallinn: Tallinn University of Technology Press, p. 125-132.

   Research Contribution: high, Paper writing contribution: high (main author)

6. Lupeikiene, A., Miliauskaite, J., Caplinskas, A. (2013). A View-based Approach to Quality of Service Modelling in Service-oriented Enterprise Systems. *In Proceedings of the 2nd International Business and Systems Conference BSC 2013*, Riga: Riga Technical University, 2013. Available at: <https://bsc2013-journals.rtu.lv/article/view/bsc.2013.2/238>

   Research Contribution: medium, Paper writing contribution: high

7. Miliauskaitė, J. (2012). Quality of Service: Concept Analysis. In Čaplinskas, A., Dzemyda, G., Lupeikienė, A., Vasilecas, O. (eds) Baltic DB&IS 2012: *Local Proceedings and Doctoral Consortium of Baltic DB&IS 2012*, CEUR Workshop Proceedings, vol. 924. Available at: http://ceur-ws.org/Vol-924/paper24.pdf (The printed version of the proceedings has been published: Databases and Information Systems. Tenth International Baltic Conference on Databases and Information Systems. Local Proceedings, Materials of Doctoral Consorcium. A. Čaplinskas, G. Dzemyda, A. Lupeikienė, O. Vasilecas (Eds.). Vilnius: Žara, 2012), 235-240.

   Research Contribution: high, Paper writing contribution: high (main author)

1.14. Outline of the Dissertation

The text of the dissertation consists of 7 main chapters, conclusions, list of references, list of publications, and appendixes. Main chapters are provided with summary and with conclusions (except Chapter 1, Chapter 2, and Chapter 3).

Chapter 1 ("Introduction") describes research context and challenges, presents the problem statement, discusses motivation, aims and objectives of the research, states research questions and hypotheses, describes research design and methods, presents research results, contributions of the dissertation, and approbation of obtained results.

Chapter 2 ("Preliminaries") offers a short introduction to define details about the terminology and concepts, used in the dissertation, introduces the basics of QoS related quality theory and the basics of fuzzy set theory and fuzzy logic.

Chapter 3 ("State of the Art") presents the critical analysis of the related works on the QoS modelling, problem fuzzification and fuzzy reasoning approaches.

Chapter 4 ("Development of the Conceptual Framework") continues the development of conceptual basis of the research. It focusses on the terms and concepts, which enable to describe in a formal way QoS_{EBS} planning problem and models, methods and algorithms for solving this problem, develops and discusses main theoretical results of the research.

Chapter 5 ("Development of Problem Fuzzification Methodology") proposes a methodology to guide problem fuzzification process, presents detailed description of the methodology steps, and demonstrates the applicability of the problem fuzzification methodology in context of QoS_{EBS}.

Chapter 6 ("Modelling and Planning of Enterprise Business Service Quality“) presents main theoretical results of the research. This chapter formalises of the QoS_{EBS} planning problem, builds problem-oriented QoS_{EBS} model, designs an ensemble of collaborating algorithms to solve the QoS_{EBS} planning problem, describes the proposed algorithms, describes the architecture and other implementation issues of the proposed QoS_{EBS} planning system.

Chapter 7 ("Experimental Research") presents results of the experimental research. This chapter provides three case studies which have been performed to demonstrate the linguistic approximation, perspective integration, and viewpoint integration.

Conclusions present the main conclusions of the dissertation.

Appendix A presents the issues of systematic literature review, review protocol, and its results. Additional appendixes containing program texts and experimental results of case studies are presented in a CD attached to the dissertation.

Chapter 2 Preliminaries

To make it easier to understand the focus of this research, it seems useful to clarify the position from which it starts. The chapter offers a very short introduction to define details about the terminology and concepts, used in the dissertation. Specifically, **Section 1** introduces the basics of QoS related quality theory and **Section 2** - the basics of fuzzy set theory and fuzzy logic.

2.1. Basics of the Quality Theory

2.1.1. Product Quality

Definition 1 Product

Product is a thing produced by labour. A tangible product is a physical object that can be perceived by touch (e.g. computer). An intangible product is a product that can only be perceived indirectly (e.g. software). Product feature is such property possessed by a product that is intended to meet customer needs.

Definition 2 Quality (According to the Merriam-Webster's Dictionary)

Quality is a general term applicable to any trait or characteristic either individual or generic. The term property implies a characteristic that belongs to a thing's essential nature and may be used to describe a type or species. The term attribute implies a quality ascribed to a thing or a being.

In the context of our research, the quality is understood in a slightly narrower sense:

> *"Quality in business, engineering and manufacturing has a pragmatic interpretation as the non-inferiority or superiority of something; it is also defined as fitness for purpose. Quality is a perceptual, conditional, and somewhat subjective attribute and may be understood differently by different people. Consumers may focus on the specification quality of a product/service, or how it compares to competitors in the marketplace. Producers might measure the*

conformance quality, or degree to which the product/service was produced correctly."[4]

However, it still remains an ill-defined and context-dependent concept. According to Garvin [54],

"Despite the interest of managers, quality remains a term that is easily misunderstood. In everyday speech, it is synonyms range from luxury and merit to excellence and value. Different companies also appear to mean different things when they use the word, as do different groups within the same firm….Scholars in four disciplines – philosophy, economics, marketing, and operations management – have explored quality, but each group has viewed it from different vantage point."

Garvin [55,56,54] has generalised different definitions of quality and described five different views on what a product quality is. This classification has been wide-accepted and discussed by many other authors, including [2,57,58,59,60,61,62]. It provides the following views:

1. **Transcendental (or metaphysical) view.** '*Quality*' is synonymous with "*innate excellence*." It is both absolute and universally recognizable, a mark of uncompromising standards and high achievements [55]. According to Pirsig,

   "…even though Quality cannot be defined, you know what it is. … Quality is neither a part of mind, nor is it a part of matter. It is a third entity which is independent of the two. ... Quality isn't a substance. Neither is it a method. It's outside of both. … It's the goal toward which method is aimed." [63]

In other words, such a concept of quality is something like Plato's form [64], the essence of Quality that exists independently of the particular things that 'participate in', an ideal, towards which we should strive, but which can never be achieved in objective reality. The best illustration of such a concept may be the so-called "ideal love" which is understood very differently by

[4] Wikipedia, https://en.wikipedia.org/wiki/Quality_(business)

different persons, depending on their nationality, culture traditions, and even their personal attitudes. According to this view, the quality of a particular product always is a *relative quality*, because it only approximates (in some degree) the ideal quality. Such a relative quality can be assessed only in qualitative terms: '*high*', '*low*', etc. Of course, such assessments are highly subjective. Even so, the transcendental view cannot be ignored, because it plays an important role in business, especially, in advertisement.

2. **Product-Based View.** Its quality is viewed from the inside perspective and defined as a precise and measurable variable. According to this view,

   > *"…differences in quality reflect differences in some ingredient or attribute possessed by a product … high quality can only be obtained at higher cost. Because quality reflects the quantity of attributes that a product contains, and because attributes are considered to be costly to produce, higher quality goods will be more expensive." [55].*

So, this view does not take into account preferences of a particular user and assumes that the absence or presence of an attribute implies a higher quality. In other words, it ignores subjective aspects of the quality. However, for competitive success, these aspects are also very important.

3. **User-Based View.** This view is based on the idea that quality is an individual matter, and things that satisfy user preferences best have the highest quality. It replaces measurable attributes of product with a user's satisfaction. According to [65],

   > ***"…**this approach is more inclusive and leads to more effective quality assertions, i.e., all product attributes are considered, and criteria are weighed to reflect user satisfaction; i.e. there is only one scale of measurement. This allows for analytically well-founded normative statements on quality. In the product-based approach, this is only possible if exactly one product attribute determines quality."*

4. **Manufacturing-Based View**. This view examines quality from the inside perspective, assumes that a product having good internal properties has also good external properties and defines quality as conformance to requirements specification, in which the requirements are stated mostly in technical terms. According to [55], in contrast to the user-based view,

   "…manufacturing-based definitions focus on the supply side of the equation, and are primarily concerned with engineering and manufacturing practices. Once a design or specification has been established, any deviation implies a reduction in quality. Excellence is equated with meeting specifications, and with 'making it right the first time'….Quality is defined in a manner that simplifies engineering and production costs. On the design site, this has led to reliability engineering, and on the manufacturing site, to an emphasis on statistical quality control. …Each of these techniques is focused on the same end: costs reduction."

So, *quality* is the degree to which a product conforms to a specification, any deviation from the specification decreases quality. Similarly to the product-based view, a manufacturing-based view defines quality in objective and measurable terms, however, focuses on making error-free products, but not on the absence or presence of some attributes. Even though it does not ignore the user's interest in quality, it assumes that this interest can be satisfied, if the product is properly constructed. The aim of "making it right the first time" is to eliminate or, at least, reduce the reworking costs. [62]. The manufacturing-based view concerns about user's needs or preferences only in the case, where they are correctly identified and reflected in the requirement specification.

5. **Value-Based View.** This view defines quality in terms of costs and prices or, in other words, as the degree of excellence at an acceptable price [66,55]. It makes a trade-off between cost and quality, that is, it concerns about providing as much quality as the customer is willing to pay for. Garvin notes that

"The difficulty in applying this approach lies in its blending of two related but distinct concepts. Quality, which is a measure of excellence, is being equated with value, which is a measure of worth. The result is a hybrid – "affordable excellence"- that lacks well-defined limits and is difficult to apply in practice." [54]

On the other hand, Boehm argues that

"…it is also hard for a value-neutral approach to provide guidance for making its products useful to people, as this involves dealing with different people's utility functions or value propositions. It is also hard to make financially responsible decisions using value-neutral methods." [67]

This position is supported by Kitchenham and Pfleeger [62], and many other authors.

Coexistence of five different and often competing views raises the following quality planning problem: "How to take into account all view and resolve conflicts among engineering, manufacturing, marketing and other departments of an enterprise in a case of a particular product?"

In [68], Juran made an effort to simplify this problem by generalising five views on quality up to two definitions:

Definition 3 Product quality (external perspective)

"Quality" means those features of products which meet customer needs and thereby provide customer satisfaction. In this sense, the meaning of quality is oriented to income. The purpose of such higher quality is to provide greater customer satisfaction and, one hopes, to increase income. However, providing more and/or better quality features usually requires an investment and hence usually involves increases in costs. Higher quality in this sense usually "costs more." [68]

Definition 4 Product quality (internal perspective)

"Quality" means freedom from deficiencies—freedom from errors that require doing work over again (rework) or that result in field failures, customer dissatisfaction, customer claims, and so on. In this sense, the meaning of quality is oriented to costs, and higher quality usually "costs less." [68]

However the problem still remains unresolved.

Software product quality provides three different views on quality.

Definition 5 Internal quality

Internal quality is the totality of characteristics of the software product from an internal view. Internal quality is measured and evaluated against the internal quality requirements. [69]

This definition is based on the manufacturing-based view.

Definition 6 External quality

External Quality is the totality of characteristics of the software product from an external view. It is the quality when the software is executed, which is typically measured and evaluated while testing in a simulated environment with simulated data using external metrics. [69]

This definition is based on the product-based view.

Definition 7 Quality in use

Quality in Use is the user's view of the quality of the software product when it is used in a specific environment and a specific context of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself. [69]

This definition is based on the user-based view.

2.1.2. Service Quality

Note that the term '*service quality*' is used speaking about economic (business) services, whereas the term '*quality of service*' (QoS) is used speaking about ICT services. Due to the great differences between various kinds of services, it is almost impossible to propose one simple definition of service [70]. In the context of our research, the most appropriate is the following definition:

Definition 8 Service

A service is any act or performance that one party can offer to another that is essentially intangible and does not result in ownership of anything. Its production may or may not be tied to a physical product. [71]

A collection of other definitions can be found in [70]. In the same source, services are classified into two classes: services consumed by persons (B2C) and that consumed by enterprises (B2B). The B2C services are divided further into services that provide products or information to the customers (e.g.

supermarket, post office, consulting) and services that individually or collectively use a resource of the service provider (e.g. cinema, public transport, repair, beauty cares, rent of an equipment). The B2B services are divided further into services that provide information (e.g. audit), services that use temporarily resources of the service provider (e.g., maintenance, training program, and rent of a machine) and logistic service (e.g. transportation).

Many authors (e.g. [72,73,74,75]) argue that even if services and products share many similarities, they also differ have a number of distinctive characteristics. They are *intangible, heterogeneous, inseparable,* and *perishable*. These characteristics are usually referred as IHIP characteristics [76]. Despite the fact that the discussion is still going on whether the IHIP characteristics are characterizing services [70], the service concept is still operationalized mainly through these characteristics [77]. The question about the applicability of IHIP characteristics to technology-based services is more complicated. For example, Moeller argues that:

> *"The characteristics of intangibility, heterogeneity, inseparability, perishability (IHIP) that have been regularly applied to services have been subjected to substantial criticism, as more and more exceptions occur. The reasons for the criticism are twofold. The focus of services marketing has changed and the development of information and communication technology has advanced dramatically." [76]*

Edvardsson et al. [77] and many other researchers also advocate that technology-based services are, in fact, storable, repeatable, often standardized and, last but not least, the service production does not involve any direct interactions with humans. On the other hand, Hofacker et al. [78] state that e-services are less tangible as traditional services, possibly, more heterogeneous, taking into account instability of hardware, software and network environment, highly flexible in terms of physical separation between consumer and producer, and can be stored indefinitely by the provider (on server disk) or user.

One more difference between products and services is that services are processes rather than things and for this reason consumer's involvement in the production of many services creates additional quality control difficulties for managers.

To define a service quality is even more difficult than to define a product quality because it is a multidimensional or multi-attribute construct [79]. Efforts to understand and identify service quality have been undertaken in the last three decades. As a result, a number of different quality models, based on the different understanding of quality, have been proposed: technical and functional quality model [80]; GAP model [81]; attribute service quality model [82]; synthesized model of service quality [83]; performance only model [84]; ideal value model of service quality [85]; evaluated performance and normed quality model [86]; IT alignment model [87]; attribute and overall affect model [88]; model of perceived service quality and satisfaction [89]; PCP attribute model [90]; retail service quality and perceived value model [91]; service quality, customer value and customer satisfaction model [92]; antecedents and mediator model [93]; internal service quality model [94]; internal service quality DEA model [95]; Internet banking model [96]; IT-based model [97]; and a model of e-service quality [98]. An exhaustive overview and analysis of all mentioned models is presented in [99]. All these models are based on the views described in section 2.1.1. A significant distinctive characteristic of many proposed models is that they differ between technical and functional service qualities. The technical quality is the quality of what a consumer actually receives as a result of his/her interaction with the service firm, and the functional quality is how he/she gets the technical outcome [80].

2.2. Basics of Fuzzy Set Theory and Fuzzy Logic

Definition 9 Fuzzy set

Let X be the universe of discourse (UoD) containing elements x. Then a set of ordered pairs $A=\{x, \mu A(x)|x \epsilon X, \mu A: X \rightarrow [0,1]\}$ *is a fuzzy set in X and,* $\mu A(x)$ *is the membership function (MF) of x in A.*

The concept of a fuzzy set is an extension of a crisp (or classical set). It describes a set without a crisp, clearly defined boundary. A fuzzy set contains elements with only a partial degree of membership. Fuzzy set A is normal (or normalized), if there exists $x_0 \in X$ with $\mu_A(x_0)=1$. The UoD X may be continuous as well as discrete.

Definition 10 Ordinary fuzzy set

An ordinary fuzzy set is a fuzzy set that is defined mathematically by assigning to each possible individual in the universe of discourse a value representing its grade of membership in the fuzzy set.

The membership functions of ordinary fuzzy sets are often overly precise. However, experts may be able to identify appropriate membership functions only approximately. Nevertheless, in this dissertation, all fuzzy sets are considered be ordinary fuzzy sets.

Definition 11 Convex fuzzy set

A fuzzy set A is convex if

$\mu_A(\lambda r_1 + (1-\lambda r_2) \geq \min(\mu_A(r_1), \mu_A(r_2))$

for all $x_1, x_2 \in X$ *and* $\lambda \in [0, 1]$.

Definition 12 Interval-valued fuzzy set

An interval-valued fuzzy set is a fuzzy set whose membership functions does not assign to each element of the universal set one real number, but a closed interval of real numbers between the identified lower and upper bounds, i.e. $A: X \to \varepsilon([0,1])$, $\varepsilon([0,1]) \subset \mathcal{P}([0,1])$.

Definition 13 Fuzzy sets of type 2

A fuzzy set of type 2 is a set of all ordinary fuzzy sets that can be defined with the universal set [0,1]. It is also called a fuzzy power set of [0,1].

The computational demands for dealing with fuzzy sets of type 2 are even greater than those for dealing with interval-valued fuzzy sets.

Definition 14 Fuzzy singleton

A fuzzy singleton is a fuzzy set with a membership function that is unit at a one particular point and zero everywhere else.

Definition 15 Fuzzification

The process of generating membership values for a fuzzy variable using membership functions.

Definition 16 Fuzzification problem

A problem how to construct a set of MFs which transform a phenomenon in question, characterized by a variable, which values are defined in the universe of discourse (UoD) and categorized according to some criterion into a linguistic variable so that the names of the given categories match the names of linguistic values. Fuzzification is a process of generating membership values for a fuzzy variable using MFs.

Definition 17 Defuzzification

Defuzzification is the process of transforming a fuzzy output of a fuzzy inference system into a crisp output.

Definition 18 Fuzzy relation

A fuzzy relation is a subset of the Cartesian product of fuzzy sets:

$R(A_1, A_2, \ldots, A_n) \subseteq A_1 \times A_2 \times \ldots \times A_n,$

where A_i (i=1, 2, …, n) are fuzzy sets.

It means that fuzzy relation $R = \{(a_1, a_2, \ldots, a_n), \mu_R: A_1 \times A_2 \times \ldots \times A_n \rightarrow [0,1]\}$.

Definition 19 Fuzzy vector

A fuzzy vector is a vector containing only the fuzzy membership values.

Example 1

Let us have a fuzzy set

$$B = \left\{\frac{0}{0} + \frac{0.2}{1} + \frac{0.3}{2} + \frac{0.5}{3} + \frac{0.7}{4} + \frac{0.9}{5} + \frac{1}{6} + \frac{0}{7} + \frac{0}{8} + \frac{0}{9} + \frac{0}{10}\right\}.$$

This set can be represented as a fuzzy vector

$$b = \{0, 0.2, 0.3, 0.5, 0.7, 0.9, 1, 0, 0, 0, 0\}.$$

█

Definition 20 Fuzzy number

A fuzzy number is a convex fuzzy set on $\mathbb{R}$ such that

(1) there exists $x \in \mathbb{R}$ with $\mu A(x) = 1$;

(2) $\mu_A(x)$ is piecewise continuous,

(3). Supp(A)=[a,b], $a \leq b$, where neither a nor b is permitted to be infinite.

A fuzzy number is a fuzzy subset in real numbers which have two properties, convexity and normality. In addition, the support of fuzzy set *A* Supp(A)=$(x \in X | \mu_A(x) > 0)$ must be a bounded interval. Note that the support of a fuzzy set is a crisp set.

Definition 21 Triangular fuzzy number

A triangular fuzzy number on $\mathbb{R}$ is a fuzzy number A which has a membership function

$$\mu_A(x) = \begin{cases} 0 & x < a_1, a_3 \le x, \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x < a_2, \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \le x < a_3, \end{cases}$$

where $a_i \in \mathbb{R}$, $i = 1, 2, 3$. It is denoted by $A = (a_1, a_2, a_3)$.

Definition 22 Fuzzy logic (in the broad sense)

Fuzzy logic in the broad sense is one of the techniques of soft-computing, i.e. computational methods tolerant to suboptimality and impreciseness (vagueness) and giving quick, simple and sufficiently good solutions. [100]

This logic is older and, better known. It is application-oriented formalism that is used mainly as a reasoning apparatus for fuzzy control, analysis of vagueness in natural language and several other application domains. The fuzzy logic in the broad sense is not asking deep logical questions and is not used for formal theoretical investigations.

Definition 23 Fuzzy logic (in the narrow sense)

Fuzzy logic (FL) in the narrow sense (both propositional and predicate logic) is a branch of many-valued logic based on the paradigm of inference under vagueness. It has the classical structure of traditional symbolic logics is a pair

$$FL = <\mathcal{J}, \mathcal{T}>,$$

where:

$\mathcal{J} = <\mathcal{C}, \mathcal{F}, \mathcal{S}, \boldsymbol{\Omega}>$ is a first-order language with

- *a set of constants $\mathcal{C}$,*
- *a set of predicate symbols $\mathcal{P}$,*
- *a set of functional symbols $\mathcal{F}$,*
- *a set of logical connectives $\mathcal{S}=\{\wedge, \vee, \Rightarrow, \neg, \otimes\}$, and*
- *a set of logical constants $\boldsymbol{\Omega}$.*

$\mathcal{T}$ is a set of graded deduction rules. It includes the generalised modus ponens.

*In the language, terms and formulas can be defined (by using the inductive principle) in the same way as for a classical first-order predicate logic. With FL, a syntactic structure is connected. It means that, using $\mathcal{T}$, for any formula ψ of this logic, it is possible to derive if that formula is provable (i.e. **truth**, in symbol $\vdash\psi$) or not. Principal tools for calculations are deduction rules which are used in the logic. Since the deduction rules are graded, we also* *receive a graded notion of a provability of ψ formula, i.e. $\vdash_{\alpha} \psi$ means that ψ is true in the logic in a degree α, where $\alpha \epsilon \Omega$. [221], [100]*

Definition 24 Linguistic label

A linguistic label is a word that expresses or identifies a fuzzy set.

Definition 25 Linguistic term

A linguistic term is a category value that can be attributed to a linguistic variable. Each linguistic term is associated with a fuzzy set, which is named by a linguistic label. For this reason, linguistic terms often are referred to as linguistic values.

Definition 26 Semantics of linguistic term

The mathematical meaning of linguistic term is represented by a fuzzy number defined on UoD.

Example 2

Triangular fuzzy numbers are described by triangular membership functions whose representation is achieved by 3-tuples (a_i, b_i, c_i), where b_i indicates the point in which the membership value is one, with a_i and c_i indicating the left and right limits of the definition domain of the membership function associated with s_i. For example,

P=(0.83.1,1), VH=(0.67,0.83,1), H=(0.5,0.67,0.83), M=(0.33,0.5,0.67), L=(0.17,0.33,0.5), VL=(0,0.17,0.33),N=(0,0,0.17),

Where P (perfect), VH (very high), M (moderate), L (low), VL (very low), and N (not acceptable) are linguistic terms.

■

Definition 27 Linguistic variable

A linguistic variable is a quintuplet (L,T(L),X,G,M), where:

- *L is the name of a linguistic variable;*
- *T(L) denotes the term set of L, i.e., the set of names of linguistic values of L, with each value being a fuzzy variable denoted generically by A and ranging across the universe of discourse X which is associated with the base variable x;*
- *X is a universe of the discourse;*
- *G is an optional syntactic rule (which usually takes the form of a grammar) for generating the names of values of L, if it is necessary; and*
- *M is a semantic rule for associating its meaning with each linguistic term of L, M(t), where $t \in T(L)$, is a fuzzy subset of A.*

A linguistic variable is a variable the values of which are words rather than numbers. It represents a concept that is measurable in some way, either objectively or subjectively (e.g., quality). Linguistic variables are characteristics of an object or situation.

Definition 28 Linguistic logic (LL)

LL is an uncertain logical system, where the truth values are fuzzy subsets with unit interval designated by the linguistic labels such as true, nearly true, undecided, nearly false, etc. The linguistic truth set of LL can be generated by a context-free grammar, with a semantic rule providing a means of computing the meaning of each linguistic truth value in LL as a fuzzy subset over [0,1] closed interval.

In general, LL is not closed under the classical logical operations of negation, conjunction, disjunction as well as implication. The result of a natural logical operation on linguistic truth values in LL would require, in general, a so called linguistic approximation of some linguistic truth value. There are three distinguished features for LL:

(i) a rule of inference whose validity is only approximate rather than being exact;

(ii) linguistic truth values expressed in linguistic terms would necessarily depend upon the semantic meaning associated with the primary truth value such as true or false, as well as their modifiers nearly, about, more or less, etc.;

(iii) truth tables now become imprecise truth tables (this is due to the difference in linguistic logic as compared to multiple valued logic, which has set valued truth-values).

Definition 29 Similarity relation

A similarity relation is a fuzzy relation $R \subseteq \mathbb{F}(X) \times \mathbb{F}(X)$, *where* $\mathbb{F}(X)=[0, 1]^X$ *is the set of all fuzzy sets on UoD X, if for all* $A, B \in F(X)$ *the following conditions hold:*

1. $R(A, A)=1$,

2. $R(A, B)=R(B, A)$,

3. $Supp(A) \cap Supp(B)=0$ *implies* $R(A, B)=0$.

Definition 30 Linguistic approximation

Let $(L,T(L),X,G,M)$ is a linguistic variable, $A_0 \subset X$ is a fuzzy set, and $R \subseteq \mathbb{F}(X) \times \mathbb{F}(X)$ is a similarity relation. Then the term $\mathbb{T} \in T(L)$, for which the similarity $R(M(\mathbb{T}), A_0)$ is maximal, is called the linguistic approximation of the fuzzy set A_0. The linguistic approximation algorithm is an algorithm which assigns $\mathbb{T}$ to A_0.

This definition is based on [101] and [5]. Shortly, the linguistic approximation algorithm is an algorithm, which assigns linguistic expression to the given fuzzy set.

Definition 31 t-norm

A function T: $[0, 1]\times[0, 1]\rightarrow[0, 1]$ is a t-norm if it satisfies the following properties:

1. $\forall a \in [0,1]\ T(a,1)=a$;
2. $\forall a,b \in [0,1]\ T(a,b)= T(b,a)$;
3. $\forall a,b,c \in [0,1]\ T(a,T(b,c))= T(T(a,b),c)$;
4. $T(a, b) \leq T(c, d)$ *whenever* $a \leq c$ *and* $b \leq d$.

Definition 32 t-conorm

A function S: $[0, 1]\times[0, 1]\rightarrow[0, 1]$ is a t-conorm if it satisfies the following properties:

1. $\forall a \in [0,1]\ S(a,0)=a$;
2. $\forall a,b \in [0,1]\ S(a,b)= S(b,a)$;
3. $\forall a,b,c \in [0,1]\ S(a,S(b,c))= S(S(a,b),c)$;
4. $S(a, b) \leq s(c, d)$ *whenever* $a \leq c$ *and* $b \leq d$.

Definition 33 Aggregation

In this dissertation, the term 'aggregation' means a procedure that combines several members of a fuzzy set into one.

Definition 34 Fuzzy aggregation

In this dissertation, the term 'fuzzy aggregation' means a procedure that reduces a set of fuzzy numbers into a unique representative (or meaningful) fuzzy number.

Definition 35 Aggregation operator

A function G: $[0, 1]\times[0, 1]\rightarrow[0, 1]$ is an aggregation operator if it satisfies the following properties:

1. $G(0,0)=0$;
2. $G(1,1)=1$;
3. $G(a, b) \leq G(c, d)$ *whenever* $a \leq c$ *and* $b \leq d$.

All t-norms and t-conorms are aggregation operators.

Definition 36 Compensative aggregation operator

An aggregation operator G is a compensative aggregation operator if it satisfies the following property:

Min (a,b)≤G(a,b)≤Max (a,b).

Compensative aggregation operators are neither conjunctive nor disjunctive operators, which compensate low values by the high values, and combine result in a medium value. They are monotonic, idempotent and are suitable for combining the values of different nature. The result of aggregation belong to interval [0,1], without any assumption about its nature. An example of a compensative aggregation operator is the arithmetic mean G(a,b)=(a+b)/2.

Note. Both fuzzy aggregation operator and compensative fuzzy aggregation operators can be extended to m-array operators for m>2.

Definition 37 Linguistic vector

In this dissertation, the term 'linguistic vector' means a vector containing only linguistic terms of a given linguistic variable. It can also be seen as a vector, which components are fuzzy numbers describing the meaning of these linguistic terms.

Chapter 3 State of the Art

The chapter presents the critical analysis of the related works on the QoS modelling, problem fuzzification and fuzzy reasoning approaches. For this aim, it explores the extent body of literature **Section 1** analyses current approaches of QoS modelling. The main aim of this analysis is to justify the hypothesis H1, H3, H8. In addition, it contributes to the development of conceptual basis of our research and presents 12 definitions of related terms. All these definitions are based on the results of the conducted bibliographic research. The aim of **Section 2** is to perform a conceptual analysis of literature sources on the current problem fuzzification approaches and to collect materials, necessary for developing the methodology to guide the QoS_{EBS} planning problem fuzzification process. The aim of **Section 3** a conceptual analysis of fuzzy reasoning formalisms and to choose such one that is best suited for inferences in the tree structures that describe the hierarchy of QoS_{EBS} properties. Finally, **Section 4** concludes the chapter.

3.1. Service-Oriented Enterprise Architecture and Quality of Business Services

Service computing [102] is a dominant applications development paradigm that, *inter allia*, suggests that business applications should be implemented in the form of services. It inherits a number of concepts and principles from earlier paradigms, first of all, from object-orientation, component-based software engineering (CBSE) [103] and open distributed processing (ODB) [104]. The most important innovation of service orientation is the manner in which the separation of concerns is done. Service-oriented architecture (SOA) is an architectural style used to implement service-oriented applications [105,106]. SOA introduces two new high-level abstractions, namely, enterprise business services and business processes. It sees an application as a set of interacting services, coordinated by a business process. In other words, SOA is “an architectural style where systems consist of service users and service providers” [107]. Service providers are those functional units of the system that offer business services. They are an analogue of servers in client-server architecture. In other words, they are software units “hosting” one or more services. It is assumed that each service provider resides in a separate computer network node accessible through a name or locator other than absolute network address. Services or, more exactly, service consumers in SOA are those functional units of the system that invoke services provided by service providers. They are an analogue of clients in client-server architecture. In other words, they are software units that form and send requests for service providers. A service consumer can dynamically discover service providers. Service providers and service consumers are role names. Each service consumer resides also in a separate computer network node. In SOA, some functional units may act in both roles (be provider or consumer). The current role depends on the existing context.

A number of different definitions of SOA exist. In our research, we adopted the following definitions given by Bieberstein et al [108]:

Definition 38 Service-oriented architecture

Service-oriented architecture (SOA) is a framework for integrating business processes and supporting IT infrastructure as secure, standardized components – services – that can be reused and combined to address changing business priorities.

The symbiosis between an Enterprise Architecture (EA) [109,110]) and Service-Oriented Architecture (SOA) results in the so-called Service-Oriented Enterprise Architecture (SoEA) [111,112,113] and brings up new problems for Service-Oriented Enterprise Systems (SoES) engineering [114].

Enterprise architecture is defined in the following way [115]:

Definition 39 Enterprise architecture

Enterprise architecture (EA) is "an aggregated, holistic view of all systems, people, and internal and external constructs that have relationships within the enterprise. Furthermore, it is bound and guided by a common requirements vision (CRV) and a set of conceptual architecture principles that guide the selection, creation, and implementation of business, information, technology, and solution future states."

SoEA is a substyle of SOA. Therefore, SoEA introduces two new high-level abstractions, namely, enterprise business services (EBS) and enterprise business processes (EBP). Enterprise business services are the abstractions of existing application capabilities, which are aligned with the enterprise business functions. Enterprise business processes are the abstractions of the overall business functioning. EBS is a mechanism by which the needs and capabilities are brought together. It is defined in the following way [116]:

Definition 40 Enterprise Business Service

Enterprise business service (EBS) is "the notional or existing business functionality that would address a well-defined need. Service is therefore the implementation of such business functionality that it is accessible through a well-defined interface".

In other words, an EBS is a unit of business logic that implements one well-defined action, for example, creates an order.

EBPs are the abstractions of the overall business functioning. It is defined in the following way [117]:

Definition 41 Enterprise Business Process

Enterprise business process (EBP) is "the end-to-end (cross-departmental, and often, cross-company) coordination of work activities that create and deliver ultimate value to customers."

In other words, a business process is

> *"…an ordering of activities with a beginning and end: it has inputs (in terms of resources, materials, and information) and a specified output (the results it produces." [118]*

In SoES, business processes play crucial role. This role is twofold: business processes is used as a management tool, which helps to organize people for greater agility, and business process, more exactly, an executable model of this process is used as a tool, which helps to organize technology for greater agility [119]. There are two kinds of business models: executable and abstract.

Definition 42 Executable enterprise business process model

An executable enterprise business process model (further, executable process) is a platform-oriented process model that specifies the execution order between a number of activities constituting the process, the partners involved in the process, the messages exchanged between these partners, and the fault and exception handling specifying the behaviour in cases of errors and exceptions. [120]

Executable business processes model actual behaviour of a participant in a business interaction. Processes are executed by the SoEA execution engine, which is referred to as Enterprise Service Bus (ESB). ESB models business activities by EBSs or their compositions.

Definition 43 Abstract enterprise business process model

An abstract enterprise business process model (or abstract process) is a platform independent model that only partially specifies this process and is not intended to be executed. The model hides some of the required for execution concrete operational details. Two mechanisms are used for hiding operational details: (1) the use of explicit opaque tokens and (2) omission. (Based on [120])

Concrete operational details are added to an abstract process deploying in onto concrete platform.

In other words, the term '*process*' in EoSA refers to linked business services and enables the coordination of distributed systems supporting business processes. Such processes should never be confused with real-life business processes [121].

Business activities are mapped to business services using the so-called service orchestration process.

Definition 44 Service orchestration process

Service orchestration process is the process of coordination and arrangement of multiple services exposed as a single aggregate service.

Definition 45 Service orchestration

Service orchestration is the result of a service orchestration process.

It means that

> *"Developers utilize service orchestration to support the automation of business processes by loosely coupling services across different applications and enterprises and creating "second-generation," composite applications. In other words, service orchestration is the combination of service interactions to create higher-level business services. "*[5]

[5] Service Orchestration: Making SOA Work. MuleSoft, https://www.mulesoft.com/resources/esb/service-orchestration-and-soa#sthash.iJSmHsd4.dpuf

Orchestration is similar to an organizational workflow but the first one is conducted in a SoES and the other, in an enterprise. So, the differences between the terms *service orchestration* and *organizational workflow* are similar to those between the *EoSA business process* and *real life business process*.

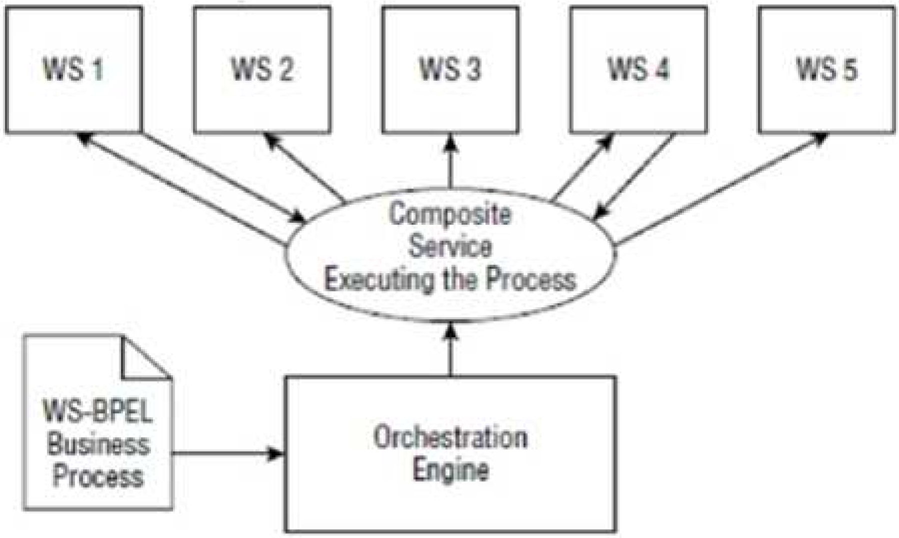


Fig. 3 Orchestration engine (Source: [122])

Definition 46 Orchestration engine

An orchestration engine (Fig. 3) is a single endpoint central process (itself implemented as a service), which coordinates the execution of different operations on the services which participate in the EoSA business process model. The invoked services neither know nor need to know that they are involved and playing a role in an EoSA business process model. Only the orchestration engine is conscious of this aim. (Based on [123,124])

There a number of different orchestration engines [125], for example, ExpressBPEL (CodeBrew technologies), BizTalk Server (Microsoft), Oracle BPL Process Manager (Oracle), WebSphere Process Server (IBM), OW2 Orchestra (OW2), etc.

In the SoES, the role of orchestration engine usually is played by Enterprise Service Bus. According to [126], it is not an easy task to define this concept:

> *"What is an Enterprise Service Bus? The question is hard to answer since there is no general consensus about a common definition of the term. There are many discussions on which features have to be*

included or which technologies should be used when realizing an Enterprise Service Bus. In contrast to that there are many vendors in the market who state that their solutions are Enterprise Service Buses or base on Enterprise Service Bus principles."

Nevertheless, it can be defined operationally, i.e. by its functional capabilities.

Definition 47 Enterprise Service Bus

Enterprise Service Bus (ESB) is a broker infrastructure which offers the following functionality:

- *services invocation;*
- *secure messaging;*
- *data transformation;*
- *adaptation of applications,*
- *process execution monitoring and controlling;*
- *orchestration;*
- *processing of complex events; and*
- *application integration tooling.*

For details, see [126]. From this definition follows that orchestration engine is only one role (and not a most important one) played by ESB. In any case, the result of orchestration is a high-order EBS that can be further used in other orchestration processes.

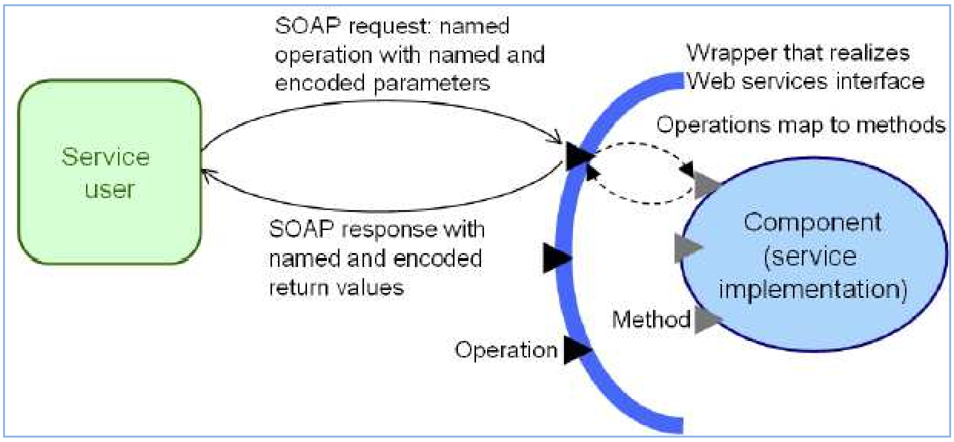


Fig. 4 SOA service (Source: [107])

To perform the orchestration, it is necessary that all EBSs have standardised interfaces and communicate via messages. In SoEA (and in SOA, too) it is achieved wrapping EBSs by web services (Fig. 4).

Definition 48 Web service

Web service (WS) is "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description" [107]).

According to this definition, web services are used to wrap the distributed components (or some legacy software) and to implement service providers' interfaces. In other words, a web service is the mechanism that converts components (or legacy software) into EBSs by wrapping EBS providers and by creating unified platform agnostic interfaces, which allow accessing EBSs via the Internet. The W3C open specification [127] defines a platform independent XML-based machine-readable interface description language – Web Services Description Language (WSDL) – which allows describing the functionality offered by a web service that wraps the component or other software. More generally,

> *"A Web service can be implemented by a concrete agent which is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided." [128]*

Web services platform also provides a number of other open specifications centred around the interface descriptions based on WSDL, web services messaging framework, and service description registration and discovery.

> *"Web services are XML-based interface technologies; they are not executable; they do not have an execution environment—they depend upon other technologies for their execution environments." [129]*

The term '*component'* is used here in a very broad sense. It may be a software component, a piece of legacy software, some hardware (e.g. printer), or even a manual procedure. However, web-based SoEA (further, webSoEA) deals only with some software. Further, we refer to this software using the term '*component*'.

There are two substyles of webSoEA: one that meets WS-* specifications[6] and other that is based on Representation State Transfer (REST) architectural style [130,131] and enterprise Web 2.0 [132]. In this dissertation, we deal only with WS-* based webSoEA. In addition, there are a number of different viewpoints how webSOA (and webSoEA) should be implemented using web services, Microsoft Windows Communication Foundation [133], IBM Websphere [134], SAP Enterprise SOA [113] among others. Despite all above mentioned differences, it is possible to ignore these differences and to discuss planning of EBS quality at the more general webSoEA level. Before starting this discussion, we summarize the most important differences between webSOA and webSoEA (Table 1)

Table 1 webSOA vs. webSoEA

webSOA	webSoEA
Internet-wide open system. Developed in a bottom-up manner.	Relatively closed enterprise-wide system controlled on an enterprise-wide level. Developed in a top-down manner. Enterprise service inventory.
Any business services. No ability to define global data types and normalize[7] business services.	Normalized enterprise business services aligned with the enterprise business functions, the use of global data types.
Not purported to support a particular business strategy and to implement predefined business processes.	Business-driven, i.e., support enterprise's business processes, strategy and objectives. Enterprise business processes coordinate compositions of interacting EBSs.
No guide on the set of services, on how they are built and deployed. No control over changes in services.	EBSs are designed, developed and deployed in compliance with the enterprise-wide standards. All changes are under enterprise's control.
The structure of messages is standardized (e.g. by SOAP) but not unified. ESB interfaces are standardized (by WSDL), but not clearly defined, not stable. No ability to use global data types in the interfaces.	The structure of messages is unified. EBS interfaces are clearly defined, stable, and make use of global data types.
Service level agreements (SLAs) are negotiated between providers and	SLAs are mandated (mostly) at the enterprise-wide system at the design

[6] See Web Services Specifications Index Page at https://msdn.microsoft.com/en-us/library/ms951274.aspx
[7] Normalisation means that each EBS should be designed with the intent to avoid similar or duplicate bodies of service logic.

webSOA	webSoEA
consumers at the run time.	time.
Direct pear-to-pear communication between consumer and provider. UDDI for service registration and discovery.	ESB as a mediator between consumers and providers.
Neither service providers nor consumers can control the SOA infrastructure and communication networks.	Intranet, extranet, and the whole infrastructure, including ESB, servers and other elements, is under control of the enterprise.
Recommended security and safety standards.	Mandatory security and safety standards.
Some services are situation-aware but only in rare cases are context-aware because the context as a rule is ill-defined.	All services are context-aware because they run in the well-defined enterprise context.

The webSoEA provides a special directory service (also referred to as service discovery agency) that allows service consumers to register and discover any EBS. Besides, any EBS is dynamically bound. It means that a service consumer does not need the EBS implementation available at build-time because the service is located and bound at runtime [107].

The webSoEA provides guidelines for creating and using service-oriented applications. SoES is business-driven, that is, it must support the enterprise's business strategy and objectives. It means that business processes in SoES must be designed keeping in mind this goal. On the other hand, business processes should be translated into abstracted and normalised EBSs drawing on global data types. Normalisation means that each EBS should be designed with the intent to avoid functional overlaps and to reduce the redundancies in EBSs, i.e., to avoid similar or duplicate bodies of service logic. Global data types are enterprise-wide defined data types based on the international standards [135].

In SoEA, EBSs have also some specifics. First of all, all EBSs, including those maintained by the external providers, should be designed, developed and deployed in compliance with enterprise-wide standards. It means that the structures of both services and messages must be unified [136]. Interfaces for all EBSs must be clearly defined and stable, and make use of global data types [112]. The enterprise business service must meet the functional objectives within the context of the business unit and the enterprise [111]. Internal directory service must be provided for registration and discovery of EBSs. This specific enables one to balance viewpoints of different EBS stakeholders.

Namely these differences motivated the scope of our research. It is obvious that the problem of planning quality of business services in the webSOA environment has quite different, probabilistic nature and cannot be solved applying methods proposed in our dissertation.

Let us discuss now the EBSs quality modelling issues.

As was already mentioned (see Section 1.1), the term *Quality of Service* (QoS) was introduced in telecommunications networks and extended later for different kinds of other ICT-based services, including Web Services, SoA services, and SoEA services. Fig. 4 demonstrates that in this new context exist a number of perspectives on service quality. Later (see Section Chapter 4) we will discuss this question in details.

In the context of webSoES, the quality of an EBS is typically addressed by the term *QoS for web services* (QoS_{WS}) that, unfortunately, causes some confusion. One of the most popular QoS_{WS} definitions is presented in [137].

Definition 49 Quality of Services for WS

Quality of Service for web service (QoS_{WS}) is a set of non-functional attributes of the entities used in the path from the WS to the client that bear on the WS's ability to satisfy stated or implied needs in an end-to-end fashion.

This definition speaks about WS end-to-end quality but it is not applicable in the context of webSoEA, because it ignores the fact that, in this context, each web service wraps some component (Fig. 4) and ignores the quality of this component as well as the specific quality requirements of a particular application domain. Using the terms proposed by Christian Grönroos [138], it speaks only about the technical quality of service and ignores its functional quality. In other words, this definition assumes that service quality and product quality are strongly separable. This assumption contradicts our hypotheses H8 and H1. However, our point of view is supported by a great number of researchers in papers on product/service continuum and IHIP characteristics (for example, [74,76,139,140,141]). Already Shostack [142] highlights the fact that the distinction between services and products is not clear cut, and that there are few pure services and products (Fig. 5). Grönroos [143] even developed a concept of the service product – the service offering – which is geared to the concept of perceived service quality. On the other hand, the Systematic Literature Review conducted preparing this chapter of the dissertation (see Appendix 0) did not discover serious arguments against hypotheses H8 and H1.

Fig. 5 Product service continuum (Source: CS Odessa corp., 2013)

In summary, the WS QoS and the QoS_{EBS} are different things.

There are also a number of QoS definitions for SOA services (QoS_{SOA}). Almost all these definitions define QoS_{SOA} through some context-dependent QoS_{SOA} model. The critical analysis of various service quality and QoS models can be found in many papers including [144,145,1,99,146]. This analysis shows that no QoS model is commonly accepted. No common accepted operational definition allowing us to measure service quality for any service exists. The majority of QoS models for services was proposed by the Web service community and describes only technical attributes. In addition, authors conclude that most of the models lack the richness needed in specifying the QoS of different types of services. Metalevel analysis conducted in [147] shows that QoS_{SOA} still remains a not well defined and frequently misused term. Besides, all existing definitions ignore multipartite and fuzzy nature of QoS_{EBS} and cannot be directly applicable in our research.

There are also several QoS_{EBS} related papers [148,149,150,151], however they focus only on some specific aspects of the problem and no one investigates QoS_{EBS} modelling and planning problem systematically. To the best our knowledge, only two papers [152,153] address the QoS_{EBS} modelling problem directly. In [152], this problem is considered from the viewpoint of the design of the whole system, i.e. this paper considers only such properties of service quality as loose coupling, composability, granularity, etc. The modelling and planning of QoS for individual services is out of scope of this work. Paper [153] focuses on the issues of the measurement and evaluation of EBS performance. Authors differ over hard and soft quality factors:

> *"...soft factors (like friendliness and competence of the employees) play an important role. The measurement and evaluation of soft factors is very challenging. Soft factors cannot be measured by using objective measuring equipment (like the measurement of throughput time with the aid of a stop watch). Soft factors rather have to be measured and evaluated by people. Here people function as subjective measuring equipments. The use of Liker scales is the common way to measure and evaluate soft factors. But Likert scales do not sufficiently consider human perception."*

The paper proposes a conceptual five stage model based on the fuzzy set theory to measure and evaluate the performance of service. However, it does not consider other than performance related factors.

Let us discuss the proposed QoS models at a more detailed level. It is not easy to say which of these models, if any, is the best suited or most representative one. We classify the proposed models into three classes: taxonomy-based models, activity-based models, and ontology-based ones.

Taxonomy-based QoS Modelling. Taxonomy-based QoS models structure a quality along the characteristics (e.g. security, interoperability, reliability, usability, efficiency, maintainability and portability). In other words, these models are more or less exhaustive taxonomies of the QoS characteristics.

A typical example of taxonomy-based model is SQuaRE based the Web Service Quality Model (SQuaRE-WSQM). This model was proposed by Abramowicz and al. [154]. It is based on the ISO/IEC Software Product Quality Requirements and Evaluation Model (SQuaRE) [69]. The authors suggest [154,155] that the web services quality model should be compatible with the software ISO/IEC 25010 (SQuaRE) model [156] because:

> *"...definition of quality requirements starts from the same set of requirements both for Web Service and software module"" [154]*.

According to their opinion, quality requirements for a service should be analogous to the ones for a software component, which produces required output. The SQuaRE-WSQM defines service quality from external, internal and quality in use perspectives. External quality is the capability of a service to provide the effects satisfying needs when this service is used under specific conditions. In other words, external quality characterises "black box" behaviour of the service. Internal quality gives a "white box" view to service quality. Both external and internal qualities are defined by the top level quality attributes: security, interoperability, reliability, usability, efficiency, maintainability, and portability. The quality in use defines quality as a utility for a specific user to achieve his/her specific goals in a specific context. It is defined by the SQuaRE model attributes used to describe the usability in use, context in use, safety in use, security in use, support in use, and adaptability in use [69]. External and internal qualities reflect the viewpoint of service owner, while quality in use, that of a service requestor. The proposed model defines three-level taxonomy of quality attributes: main (or top-level) attributes, lower level attributes, and quality measures. However, the SQuaRE-WSQM ignores the specific of services (comparing to software products) as well as service related business issues. This is the main shortcoming of all taxonomy-based QoS models. Besides, the classification schemes used in taxonomy-based models often lack clear semantics for relationships between supperlevel and sublevel quality attributes [157].

Activity-based QoS Modelling. Originally the activity-based quality models were introduced with the aim to model software maintainability [157]. Later they were adapted to QoS modelling. In activity-based QoS models, the quality is described along the activities performed on or with a SOA system. It means that in such models quality concerns are separated by activities. This approach attempts to remove the main shortcoming of taxonomy-based approach and to take into account the specific of services. This attempt has been made by the OASIS Committee [158,159]. These documents emphasise that web services differ from installation-based software. The differences cause a distinct web service quality model and attributes. First of all, service consumer and provider as a rule belong to different ownership domains and relationships between their instances can be established ad-hoc. This includes a possibility of a web client to dynamically change the server. The changes can also be done in real time when quality is not sufficient. Secondly, the quality of web services depends on the runtime environment. Consequently, variation of service quality can occur. Thirdly, service consumer must tolerate some acceptable deviation of required quality because it may be not obtainable.

The OASIS Web Service Quality Model [158] consists of three components: quality factors, quality associates, and quality activities. A quality factor is a group of attributes, which represent web service's properties. In OASIS terminology, the term "quality factor" is broader than the term "quality attribute" in its common usage. A quality associate is a person or organisation (in other words, a role) related to web services life cycle stages. The quality activity refers to various actions performed by associates to ensure web services quality and its stability. OASIS Specification emphasises that the quality model should be established from a service but not from a product viewpoint. It implies different views of using a service. So, quality can be considered from different perspectives: user's perspective, interoperability perspective, and management and security perspective [158].

Quality factors are divided into two groups: business quality factors and system quality factors. Business quality factors enable evaluating the business value of services, i.e. the economic worth delivered by applying these services on a business. The business value depends on quality subfactors such as price, penalty and incentive, business performance, service recognition, service reputation, and service provider reputation. In addition to those factors, business benefit, profit and return of investment can be included in this group of factors. System quality factors are divided into variant quality part and invariant factors. The values of variant factors can be dynamically varied in run-time, while the values of invariant factors should be determined immediately after the service development process is completed. Invariant factors include interoperability, business processing quality, manageability and security. Values of response time, maximum throughput, availability, accessibility, and successability vary dynamically.

The main shortcomings of the OASIS Web Service Model are that it ignores domain specific nature of some quality attributes and provides only three views of quality, which is not enough. Thus, there is a need to develop such approach, which provides the integration of all viewpoints and perspectives on service quality at a higher abstraction level. This conclusion is in line with the aims of our research. On the other hand, this research to some extent was inspired by the philosophy beyond this OASIS Web Service model.

Goal-oriented methodology. Closely related to our research also are works on the goal-oriented methodology [160] and the application of the i* framework [24] for the enterprise [161,162] and software requirements [163,164] modelling and reasoning about the software quality. The goal-oriented methodology inspired our ideas about modelling interdependencies between viewpoints and between perspectives. This methodology and works on the i^* framework made also a strong impact on our approach to the formalisation of QoS models and reasoning procedures about alternative configurations of QoS attributes' values in order to compromise requirements stated on the basis of different viewpoints.

Pioneering works on the design of automated reasoning procedures for the i^* framework were published by Giorgini et al [165,166]. These ideas were developed further in [167,144,168]. The authors proposed a number of qualitative and quantitative procedures for goal model analysis, which separately propagate negative and positive evidence, are fully automated, and work in a forwards and backwards direction. The proposed algorithms are sound and complete. The algorithms take as input labels for some of the lower goals of the model and infer other labels higher up. In other words, given a formal axiomatic goal model and labels for some of the goals, the algorithms propagate these labels forward towards root goals. If the graph contains loops, this is done until a fix point is reached. An axiomatization of goal models also was proposed by Giorgini and his colleagues [169,170]. However, the majority of these procedures emphasize automated reasoning over goal models ignoring the interactive nature of such analysis. As is pointed out in [171],

> *"The full automation in these procedures does not give the evaluator freedom to make decisions in the presence of conflicting, partial or unknown information"*.

An interactive qualitative approach allows one to narrow the number of alternatives and to further test the feasibility of remained alternatives using some automated quantitative procedures. Stirna and Persson [162] developed one of the first procedures of this type. It was developed as a part of the NFR Framework [172] and was based on the notion of goal "satisficing". The procedure pretended to be extensible for the i^* framework, however, it has emerged that its interactivity level is too restrictive to be effective applied to i^* models. Horkoff and Yu [171,26] developed other interactive qualitative procedure for goal- and agent-oriented models. This procedure is applicable to i^* models and allows an evaluator to compare alternatives in the domain by asking "what if?" type questions. It could also be applied to the NFR Framework [168] and GRL [173] because both these approaches are syntactic subsets of the i^* framework. The process starts by assigning initial values to labels expressing the degree of satisfaction or denial to intentions related to the analysis question. Using the preliminary defined rules, these values are propagated through the model links. Human judgement is required in cases when it is necessary to combine multiple conflicting or partial values. An evaluator analyses the final values taking into account the original “what if?” question. This idea is also was approved to be used in our research.

Despite all the above mentioned advantages and the fact that goal-oriented methodology deals with so called ‘*soft goals*’, it cannot be directly applied to solve the preliminary planning of QoS_{EBS} problem (see sections 1.2 and 6.1). From the point of view of our research, the main shortcoming of the goal-oriented methodology is that its reasoning engine is not suitable to perform reasoning in a linguistic logic.

Summarising the results of the bibliographic research done in this section, we can claim, that any of the analysed QoS modelling approaches cannot model QoS_{EBS} in such a way that an effective ensemble of collaborative methods to resolve conflicts among the requirements stated by different stakeholders is possible to design. In other words, this bibliographic research supports our hypothesis H3.

3.2. Problem Fuzzification Approaches

The proper MFs construction methods are very important in the modelling of QoS_{EBS} and should be chosen very carefully. There exist a plethora of various approaches to the constructing of MFs, (e.g. [174,175,176,177]). A decision which approach should be used to construct MFs in a particular case depends on many circumstances. The most fundamental one among them is the chosen semantic of a fuzzy set, which, in turn, depends greatly on the problem in question. Three main semantics of MF are the modelling of similarity (imprecision), preference (vagueness), and uncertainty [178]. We interpret fuzziness as vagueness or, in other words, use fuzzy sets to model vague human concepts. Despite the chosen understanding of fuzziness, all MF construction approaches can technically be divided into the *manual* and *automatic* ones (Fig. 6).

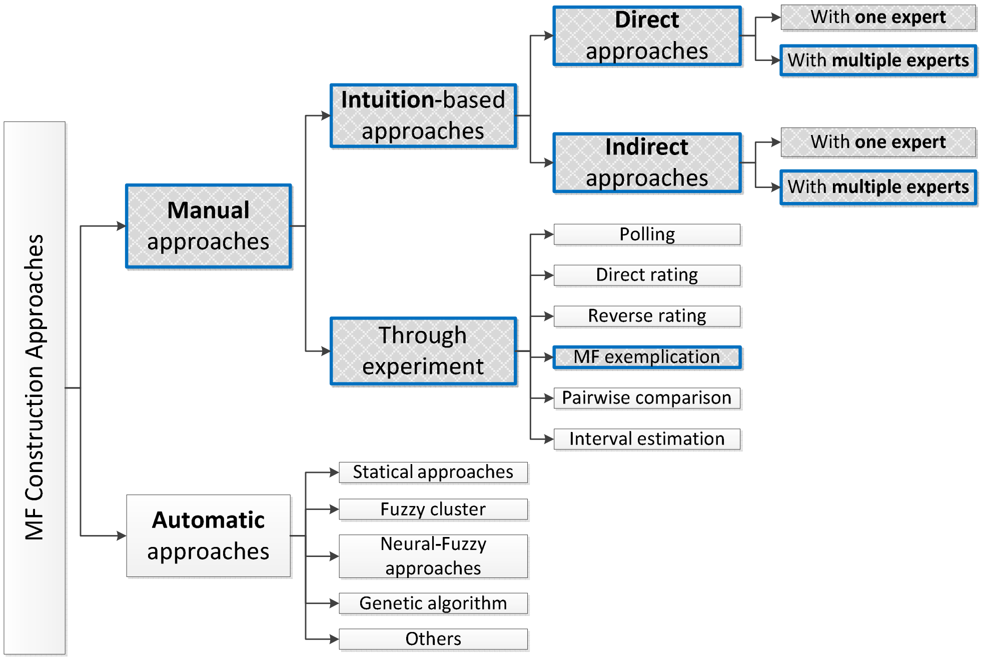


Fig. 6 MF construction approaches

The main characteristics of *manual approaches* are: 1) the usage of the empirical data, typically collected applying some phenomenography-based methodology; 2) a relatively small amount of collected data; 3) monotonous, time-consuming, and less efficient than an automatic MF construction process; 4) a possible subjective bias caused by the improper selection of interviewees; 5) a possible problem-related bias caused by inappropriate knowledge acquisition techniques [179].

The main characteristics of *automatic approaches* are: 1) nonattendance of experts, 2) supplied large data sets that are used to extract knowledge about the shape and parameters under consideration; 3) non-transparency (any justification of the result); 4) adjusting MF through learning, optimization, or using other techniques. The supplied data sets are often graphically represented in a normalized relative frequency function and histograms [180]. They contain samples of MF values for some elements of the fuzzy set under construction. The automatic approaches are adaptive in the sense that they generate initial MF from the supplied data set and further adaptively change this MF when additional data sets are provided. It means that these approaches can also be used in the cases where MF should be changed dynamically in real-time systems.

The manual MF construction approaches can be further subdivided into:

- *Intuition-based MF construction approaches.* In these approaches, the shape and parameters of MF are defined processing phenomenographic descriptions, prepared by experts in the field (interviewees) on the basis of their subjective perceiving of the quality. Usually, their understanding of the quality depends not only on their personal attitudes, but also on their individual knowledge, innate intelligence, experience, and, possibly, on the relevant literature. The final decision on the shape and parameters of MF under consideration is made by its developer (interviewer) on the basis of experts' opinions as well as on the basis of his subjective judgement.

- *MF construction through experiments*. These approaches rely on psycholinguistic experiments by which the MF developer investigates what the given linguistic terms "mean" to the experts who represent different understandings of the quality. The experiments can be carried out using different assumptions on the nature of fuzziness (e.g. interpersonal disagreement or individual subjective uncertainty) and applying different techniques (e.g. rating, exemplification, interval estimation, etc.).

The automatic MF construction approaches can be further subdivided as follows [176]:

- *Statistical approaches*. There exists a great number of various statistical approaches (e.g. histogram-based methods [181], frequency-driven [182], etc.) that combine various statistical techniques in different ways. One among them was proposed in [183]. It is a simple method that maintains MF understandability. MF is constructed by applying statistical techniques to calculate MF centres, spread, overlap, slope, etc. The method helps to provide initial intervals that define linguistic variables, and to identify the optimal parameters for MFs. A general shortcoming of statistical approaches is a questionable reliability of statistical data because such data can be biased on spot noises.

- *Fuzzy cluster*. Clusters can be treated as subsets of a supplied data set. Consequently, they can be classified as crisp (hard) or fuzzy (soft) clusters [184]. In fuzzy clustering, data elements can belong to more than one cluster. So, each data element can be associated with a set of membership levels. For details of the fuzzy clustering-based MF construction procedure see in [185,186].
- *Neuro-fuzzy approaches*. There exist several neuro-fuzzy techniques used for the MF construction [175]. All these techniques are based on the integration of artificial neural networks and fuzzy sets theory. The main idea is to use some neuro-fuzzy learning algorithm [187]for adjusting the parameters of MF, extracted from the supplied data sets. Inter alia, this approach allows us to construct dynamical MF that is dependent on the available values of variables at a given time moment t [188].
- *Genetic algorithms*. In the MF construction process genetic algorithms are used to cluster the values of quantitative attributes into fuzzy sets with respect to the given fitness evaluation criteria. Many different algorithms (e.g. [189,190,191]) were proposed for this aim. They differ in fitness functions, chromosome encoding, selection procedures, and other details.
- *Others*. The most important automatic approaches for constructing MF include other methods, such as inductive reasoning [192], deformable prototypes [193], gradient methods [194], etc.

Intuition-based approaches can be subdivided further into direct and indirect ones. In either direct or indirect approach, single or multiple experts' opinions can be taken into account [195,196]. The main characteristics of direct approaches are: 1) assumption that vagueness arises from an individual subjective uncertainty; 2) MF is constructed using some aggregation technique of (possible, weighted) experts' evaluations (i.e. degree of membership), assigned to the given crisp values, mapped to a fuzzy set under construction (instead of aggregation, some interpolation technique can be used); 3) used to fuzzify the concepts with measurable properties (e.g. execution time or throughput); 4) MF reflects subjective experts' evaluations directly (i.e. explicitly); 5) experts are required to give overly precise answers; 6) it is simple and easy to implement. The main characteristics of indirect approaches are: 1) MF is constructed on the basis of expert evaluations of certain relations (e.g. pair-wise comparisons) among the elements within the crisp set under consideration; 2) MF reflects subjective experts' evaluations indirectly (i.e. implicitly); 3) less sensitive to various biases of subjective judgment.

MF construction approaches through the experiment ([174,177,197] can be further subdivided into:

- *Polling*. It is assumed that the fuzziness arises from interpersonal disagreements. The different experts answer the question: "Do you agree that object/subject is a linguistic term *F*?" The answers of yes/no type are polled and the average is taken to construct MF.
- *Direct rating*. It is assumed that the fuzziness arises from individual subjective vagueness. The same question "How *F* is *a*?" is given to the same expert over and over again, and the answers are compared to that MF, predefined by the experimenter. The construction of MF is based on the frequency of a particular response.
- *Reverse rating*. It is assumed that fuzziness arises from individual subjective vagueness. The expert, who defines MF, is asked to indicate how much strongly a given crisp value under evaluation corresponds to the given linguistic term. This approach can be used for periodical verification of the results obtained by the direct rating method.
- *MF exemplication* (also called continuous direct evaluation). Experts are asked the question "To what degree does a given crisp value belong to the linguistic term *F*?" and to express the compatibility of each term

with each combination of items by answering yes/no and assigning the numbers from 0 to n to indicate their degree of confidence in the answer. A great variability of answers is likely. The approach is oriented to the trained experts.

- *Pairwise comparison*. Experts are asked to select an object that explains the fuzzy variable best from a pair of objects. The question is: "Which is more *F* (by how much)?" MF is constructed combining the results.
- *Interval estimation*. Experts are asked to give an interval of crisp values that describe the linguistic term *F*. The method is appropriate to situations where a strong linear order can be defined on the measurements of the fuzzy concept.

The summary of the approaches, which are relevant to our research, described in related works, is presented in Fig. 6. In this figure, the approaches, which are relevant to our research, are outlined by thick blue lines.

The main conclusion of this section is that the membership function construction method depends on a specific of a particular EBS. No particular method is applicable to any EBS. Therefore it is necessary to develop a problem-independent methodology to guide the QoS_{EBS} planning problem fuzzification process. This section motivates also our research objective 2 (see Section 1.4).

3.3. Fuzzy Reasoning Approaches

Semantic issues. Usually, approximate reasoning is defined as an inference of a possibly imprecise conclusion from possibly imprecise premises [198]. There the term '*inference*' can be understood in several different ways:

a. as a "common sense" reasoning strategy that, typically, is based on a number of heuristics [199], [200], [201];
b. as a fuzzy reasoning strategy that deals with possibly imprecise sentences and is based on fuzzy sets and fuzzy logic [202], [203], [204], [205];

c. as a default reasoning strategy that is based on default logic, non-monotonic logic and circumscription [206], [207], [208]; and
d. as an analogical (or case-based) reasoning strategy that aims derives conclusions according to analogies to similar situations [209], [210], [211], [212].

These approaches can also be combined each with others, for example, the case-based reasoning can be integrated with the rule-based reasoning [213]. In their later publication, Dutta and Bonissone argue [214] that:

> *"The task of a reasoning system is to determine the truth value of statements describing the state or the behaviour of a real world system. However, this hypothesis evaluation requires complete and certain information, which is typically not available. Therefore, approximate reasoning techniques are used to determine a set of possible worlds that are logically consistent with the available information. These possible worlds are characterized by a set of propositional variables and their associated values. As it is generally impractical to describe these possible worlds to an acceptable level of detail, approximate reasoning techniques seek to determine some properties of the set of possible solutions or some constraints on the values of such properties."*

Ruspini supports this point of view and argues that possible word semantics is also most suitable to describe semantics of fuzzy reasoning systems [215]. According to him, in fuzzy reasoning systems,

> *"Resemblance between possible worlds is quantified by a generalized similarity relation, i.e., a function that assigns a number between 0 and 1 to every pair of possible worlds…. If the typical reasoning problem is thought of as the determination of the truth value of a proposition (the hypothesis), then an approximate reasoning problem may be described as one where available evidence does not permit such evaluation without ambiguity."*

On the other hand, Zadeh argues [216] that the meaning of linguistic term is defined by a fuzzy subset of UoD, which, in turn, is defined by the membership function of this term. More precisely, for a given linguistic term *t* (i.e. for linguistic value *t*) the function $\mu_N(t, x)$ ($x \in UoD$, N is a naming relation) defines a fuzzy subset of UoD whose membership function $\mu_t(x) = \mu_N(t, x)$. This fuzzy subset, denoted by *M(t)*, is defined to be the meaning of the term t. Equivalently, the term *t* may be viewed as a label for a fuzzy subset of UoD which "comprises" (in a fuzzy sense) those elements of UoD which are described by *t* [216]. In short, the meaning *M(t)* of linguistic term *t* is a fuzzy number defined by $\mu_N(t, x)$, $t \in T$, $x \in UoD$, where T is the set of linguistic terms and $\mu_N(t, x)$ is the membership function of the naming relation N.

Thus, a linguistic variable can also "*…be regarded either as a fuzzy number or as a variable whose values are defined in linguistic terms*" [217].

Reasoning in fuzzy logic. Fuzzy logic can be described shortly in the following way:

> *"As a mathematical object, fuzzy logic has the classical structure of a logic, i.e. it consists of a first-order language J which consists (as classically) of a set of predicate symbols* $P \in \mathcal{F}$*, a set of functional symbols* $f \in \mathcal{F}$ *and a set of logical connectives* $\{\wedge, \vee, \Rightarrow, \neg, \otimes\}$*. Moreover, J also contains a set Ω of logical constants. In that language, terms and formulas can be defined (by using the inductive principle) in the same way as for a classical first-order predicate logic. With any classical logic, a syntactic structure is connected. It means that, for any formula ψ of a logic, we can derive if that formula is provable (i.e. truth) in that logic (in symbol* $\vdash\psi$*) or not. Principal tools for calculations are deduction rules which are used in the logic. In a fuzzy logic, graded versions of deduction rules are used, and it means that we also receive a graded notion of a provability of ψ formula, i.e.* $\vdash_\alpha \psi$ *means that ψ is true in the logic in a degree α, where* $\alpha \in \Omega$*." [218]*

Most important rules for reasoning in fuzzy logic are the *generalised modus ponens* (GMP) [219] and the generalized modus tollens (GMT) [220]. GMP is a forward data-driven inference while GMT is a backward goal-driven inference. To solve our problem, we need GMP. GMP generalizes the corresponding classical rule of inference

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \qquad\quad q \end{array}$$

to its approximate version

$$\begin{array}{l} p' \\ p \rightarrow q \\ \hline \qquad\quad q' \end{array}$$

The implementation of a generalised GMP inference scheme leads to the problem of selection suitable fuzzy implication. There are over 40 different forms of implication relation reported in the literature. Table 2 presents most important of them.

Table 2 Fuzzy implication operators

Name	Formula
Zadeh	$\mu_R(x,y) = max\{min[x,y], 1-x\}$
Mamdani	$x \rightarrow y = \min\{x,y\}$
Larsen	$x \rightarrow y = xy$
Largest	$x \rightarrow y = \begin{cases} 0 & if\ x = 1\ and\ y = 0 \\ 1 & otherwise \end{cases}$
Łukasiewicz	$x \rightarrow y = \min\{1, 1-x+y\}$
Gödel	$x \rightarrow y = \begin{cases} 1 & if\ x \leq y \\ y & if\ x > y \end{cases}$
Goguen	$x \rightarrow y = \begin{cases} 1 & if\ x \leq y \\ \frac{y}{x} & if\ x > y \end{cases}$

The implications are classified according to their features into three families: t-norm implications, R-implications and S-implications (see Table 3).

Table 3 Choice of implication families

<table>
<tr><th></th><th>t-norm Implications</th><th>R-implication</th><th>S-implication</th></tr>
<tr><td rowspan="2">Examples</td><td rowspan="2">Mamdani, Larsen, etc.</td><td>Gödel, Goguen, Gaines, etc.</td><td>Diene, Dubois-Prade, Mizumoto, etc.</td></tr>
<tr><td colspan="2">Lukasiewicz</td></tr>
<tr><td rowspan="5">Features</td><td>identically under certain conditions</td><td colspan="2" rowspan="2">Slow and less precision</td></tr>
<tr><td>rapidity and precision</td></tr>
<tr><td>faster dynamics</td><td colspan="2">less dynamic</td></tr>
<tr><td>possible to cancel any static error</td><td colspan="2">cause the creation of a steady state error</td></tr>
<tr><td>Robust behaviour (Mamdani, Larsen)</td><td>Not robust behaviour</td><td>Not robust behaviour</td></tr>
</table>

The choice of most suitable implication depends on a problem or, in other words, is *context-dependent*. Today, the fuzzy reasoning systems typically are used to solve various diagnostic problems, including design of fuzzy controllers. However the requirements for fuzzy reasoning formalisms suitable to solve such diagnostic problems and formalisms suitable to solve QoS_{EBS} planning problem are quietly different (see Table 4).

Table 4 Requirements for fuzzy formalism

QoS_{EBS} planning problem	Diagnostic problem
Static environment	Dynamic environment
Prediction	Simulation or control
Expert evaluations of intervals for linguistic terms (more subjective)	Measured data values (more objective)
Accuracy of expert evaluations: moderate	Measurement accuracy: high
Computational cost (speed and memory): not so important	Computational cost (speed and memory): very important
Input: set of initial values of linguistic variables	Input: crisp value
Fuzzy reasoning	Fuzzy sets and fuzzy if-then rule based decision making process

It is important that GMP can be confronted with the triangular fuzzy numbers [221]. It is possible because, according to [218],

> *"There is another tool for verification of a provability $\vdash\psi$. Instead of syntactic methods (i.e. formal rules for handling with formulas), we can use semantic methods, i.e. methods based on interpretations*

of formulas in models. A model $\mathcal{C}$ of logic J (not important if a logic is a fuzzy logic or classical one) is based on some concrete structure (set, category, group, topological spaces, etc.) and interpretations of predicate and functional symbols in that structure."

Thus, we can interpret linguistic terms by fuzzy numbers and to apply the semantic implementation of the GMP for fuzzy reasoning. However, in this case GMP should satisfy some rational properties [222]: basic property, total indeterminance property, and subset property. Although it was found that that Mamdani implication and Larsen implication are best suited for fuzzy reasoning using GMP, working with fuzzy numbers Larsen implication is best one [223]. So, we choose namely this implication.

One more important question is related to the fact that we should perform fuzzy reasoning in hierarchies of QoS_{EBS} quality attributes. These hierarchies are represented by fuzzy AND trees. It means that in each node we have a number of GMB connected by the connective *'AND'*. This connective can be implemented as a fuzzy conjunction, which is in general associated with triangular norms [224]. However, as noted in [225], in some cases it is possible (or even desirable) to leave the domain of triangular norms and co-norms and get fuzzy aggregation operators, for example, arithmetic or geometric means. So, in our case, we use fuzzy arithmetic mean.

3.4. Conclusions

The main conclusions of this chapter are as follows:

1. The bibliographic research of literature on webSOA, webSoEA and EBS highlighted the differences between webSOA and webSoEA, and supported our decision to limit the scope of research considering only the quality of SoES business services because the problem of planning quality of EBS in webSOA environment has quietly different, probabilistic nature and cannot be solved applying methods proposed in our dissertation.
2. This research did also discover no evidences that contradict to our hypotheses H1 and H3. No one surveyed source argues against the claim

that services quality can be understood differently than product quality. We did also not find a source that proposes such modelling approach which is suitable to model QoS_{EBS} in such a way that an ensemble of collaborative algorithms to resolve conflicts among requirements stated by different stakeholders be possible to design.

3. The research also supported our hypothesis H8 that product/services continuum exists and that evaluating service quality it is not possible to ignore the quality of software product which generates results delivered by this service.
4. The bibliographic research of literature on problem fuzzification approaches shown that QoS_{EBS} planning problem fuzzification method depends on a specific of a particular EBS. No one concrete fuzzification method is suitable for any EBS. Therefore it is necessary to develop a problem-independent fuzzification methodology that can be used to guide the QoS_{EBS} planning problem fuzzification process for any EBS. Thus the research supports our hypotheses H5.
5. The bibliographic research of literature on fuzzy reasoning approaches shown that most of current fuzzy reasoning approaches are applicable only to fuzzy controllers and various diagnostic problems. In such context, the reasoning is used to classify a given UoD value as belonging to a particular linguistic term. We conclude that in order to reason about linguistic terms in tree structures, which describe the hierarchy of QoS_{EBS} attributes, it is necessary to combine fuzzy implications and semantic derivation techniques. By semantic derivation we mean the computation with fuzzy numbers.

Chapter 4 Development of the Conceptual Framework

The chapter continues the development of conceptual basis of our research. It focuses on the terms and concepts, which enable to describe in a formal way QoS_{EBS} planning problem and models, methods and algorithms for solving this problem.

Let we start now with the definition of terms and concepts directly related to the modelling of views, viewpoints, perspectives and QoS_{EBS}. The semantics of presented definitions is described in terms of set theoretic semantics. Set theoretic semantics is a kind of referential theory of meaning [226], in which the meaning of a particular term is regarded as a pointer to the designated object in the real world. In other words, the meaning of a term is what it refers to [227]. In an analogous way is defined the semantics of functional symbols, predicate symbols, terms' constructors, etc.

In the first order predicate logic formalism, the quantifiers $\exists$ and $\forall$ are "unrestricted" in the sense that $(\exists x)P(x)$ means that there is some entity in the universe of discourse *U* which has the property *P*. $(\forall x)P(x)$ means that all entities in the universe of discourse *U* have the property *P*. In this formalism the sentence "*Each element of a set A has a property P*" should be described by the formula $(\forall x)\big(A(x) \Rightarrow P(x)\big)$ and the sentence "*Some element of a set A has a property P*" should be described by the formula $(\exists x)\big(A(x) \& P(x)\big)$. In this dissertation, the quantifiers $\exists$ and $\forall$ are "restricted" and have the following set theoretic semantics:

$$\begin{aligned}
&(\forall x{:}\,A)P(x) \equiv_{def} A \subseteq P,\\
&(\exists x{:}\,A)P(x) \equiv_{def} A \cap P \neq \emptyset, \qquad (1)\\
&(\neg x{:}\,A)P(x) \equiv_{def} A \cap P \neq \emptyset.
\end{aligned}$$

In the cases where a quantifier is restricted to range only a finite set $A = \{a_i \mid 1 \leq i \leq N,\ N \in \aleph\}$ we will write $\exists_A$ and $\forall_A$. These quantifiers have the following semantics:

$$(\forall_A x)P(x) \equiv_{def} \bigcup_{i=1}^{N} \{P(a_i)\} \tag{2}$$

$$(\exists_A x)P(x) \equiv_{def} \bigcap_{i=1}^{N} \{P(a_i)\} \tag{3}$$

$$(\neg_A x)P(x) \equiv_{def} \bigcup_{i=1}^{N} \{\neg P(a_i)\} \tag{4}$$

Further, let $A = \{a_i \mid 1 \leq i \leq N,\ N \in \aleph\}$ be a finite set of elements, $<_A \subseteq A \times A$ be a partial order relation on A, $\check{R} = \{\check{r}_i \mid 0 \leq i \leq N_{\check{R}}, N_{\check{R}} \in \aleph\}$ be a finite set of numbers called *ranks*, $F^{"}: A \to \check{R}$ be a ranking function which for each element of A assigns a unique rank, i.e. $(\forall_A x)(\exists_{\check{R}} \check{r}')(F^{"}(x) = \check{r}')\&\forall_{\check{R}} \check{r} \left((F^{"}(x) = \check{r}) \Rightarrow (\check{r} = \check{r}')\right)$. Several elements of A can be of the same rank, all the elements of A with the same rank $\check{r}$ form a subset denoted by $A^{(\check{r})} = \{x \mid x \in A, F^{"}(x) = \check{r}\}$. The dependencies between the relation $<_A$ and the ranking function $F^{"}$ are described by the following formulas:

$$(\forall_A x, y)(\forall_{\check{R}} \check{r}) \Big((\check{r} > 0)\&\big(F^{"}(x) = \check{r} = 1\big) \Rightarrow$$

$$(\exists_A x_1) \Big(\big(F^{"}(x_1) = \check{r}\big)\& \Big(<_A (x_1, x)\&(\forall_A x_2)\big(F^{"}(x_2) = \check{r} - 1\big)\&(x_2 \neq x_1) \Rightarrow$$

$$\neg <_A (x_1, x_2)\Big)\Big)\Big),$$

$$(\forall_A x, y) \left(\left(x \in A^{(\check{r})}\right)\&\left(y \in A^{(\check{r})}\right) \Rightarrow \neg <_A (x, y)\&\neg <_A (x, y)\right),$$

$$(\forall_A x, y) \left(\left(x \in A^{(0)}\right) \Rightarrow \neg <_A (x, y)\right). \tag{5}$$

It means that in the set A any element x with the rank $\check{r} > 0$ has an element with the rank $\check{r} - 1$ which is greater than x with respect to the partial order relation $<_A$, two elements of the same rank are not comparable to each other with respect to the relation $<_A$, and elements with rank 0 are maximal with respect to the relation $<_A$.

Definition 50 Ranked set

$A' =_{def} \langle A, <_A, F^{"} \rangle$ is a ranked set produced by the relation $<_A$ and the ranking function $F^{"}$ on a finite set A.

Definition 51 Single-rooted tree on a ranked set

If $T \subseteq A$ and

$$(\exists_T x_1)\Big(\big(x_1 \in A^{(0)}\big)\&(\forall_T x)\Big((x \neq x_1) \Rightarrow \neg\big(x \in A^{(0)}\big)\Big)\Big),$$

$$(\forall_T x)\Bigg(\Big(\big(x \in A^{(i)}\big)\&(i > 0)\Big) \Rightarrow (\exists_T x_1)\Big(\big(x_1 \in A^{(i-1)}\big)\& <_A (x, x_1)\&(\forall_T y)\Big((y \neq x_1)\&\big(y \in A^{(i-1)}\big)\Big) \Rightarrow \neg<_A (y, x_1)\Big)\Bigg) \qquad (6)$$

then $T' =_{def} \langle T, <_A, F^{"} \rangle$ is a connected acyclic single-rooted tree on a ranked set A' with the set of nodes T, the set of edges $E = \{(x,y) \mid x \in T, y \in T, F^{"}(x) = i-1, F^{"}(y) = i, <_A (y,x)\}$, the single root $x_1 \in T \cap A^{(0)}$, and the set of terminal nodes (leaves) $L = \{x \mid x \in T, (\forall_T y)\neg<_A (x,y)\}$.

Let $x \in \{T - L\}$ be a node of tree T' for which $F^{"}(x) = i, 0 \leq i \leq N - 1$. The elements of the set $Ch_x\{y \mid y \in T, F^{"}(y) = i+1, <_A (y, x_1)\}$ are called *children* of the node x and the node x is called *parent* of the nodes y.

Let $T_{and} \subset T, T_{or} \subset T, T_{and} \cap T_{or} = \emptyset, T_{and} \cup T_{or} \cup L = T, |T_{and}| = n, |T_{or}| = m,\ m, n \in \aleph$ be a partition that divides a set of nodes of tree T into three disjoint subsets: set of AND nodes T_{and}, set of OR nodes T_{or} and set of leaf nodes L. Let P be some property defined on the set T. In a particular node $t \in T$ this property can be *satisfied* or *unsatisfied*. We will write $P(t)$, if the property is satisfied at the node t and $\neg P(x)$, otherwise. In other words, P is a linguistic variable with a set of linguistic terms $P_{tr} = \{satisfied, unsatisfied\}$ and with the following semantics:

$$P = \big\{P(t), \mu_P(t) \mid t \in T, P(t) \in P_{tr}, \mu_P: T \rightarrow \{0,1\}\big\}, \qquad (7)$$

where μ_P is a membership function. The semantics of the AND and OR nodes is defined by the following formulas:

$$(\forall_T x: T_{and})\Big(\big((\forall_T y: Ch_x)P(y) \Rightarrow P(x)\big)\&\big((\exists_T y: Ch_x)\neg P(y) \Rightarrow \neg P(x)\big)\Big),$$

$$(\forall_T x: T_{or})\Big(\big((\exists_T y: Ch_x)P(y) \Rightarrow P(x)\big)\&\big((\forall_T y: Ch_x)\neg P(y) \Rightarrow P(x)\big)\Big). \qquad (8)$$

It means that at the node AND the property P is *satisfied*, iff it is *satisfied* at all the children nodes and that this property at the OR node is *satisfied*, iff it is *satisfied* at least at one children node. We define an implication relation $r^{(P)} \subset T^N$ which at each node $t \in T_{and} \cup T_{or}$ infers in the forward manner the value of property $P$from the values of P of its child nodes, i.e.

$(\forall_T x: T_{and})\left((\forall_T y: Ch_x)P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow P(x)\right)$,

$(\forall_T x: T_{or})\left((\exists_T y: Ch_x)P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow P(x)\right)$,

$(\forall_T x: T_{and})\left((\exists_T y: Ch_x)\neg P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow \neg P(x)\right)$,

$(\forall_T x: T_{or})\left((\forall_T y: Ch_x)\neg P(y) \& r^{(P)}(x, y_1, \dots, y_n) \Rightarrow \neg P(x)\right)$.

This inference is based on the *modus ponens* rule.

Definition 52 AND/OR tree

$T^{(P)}_{and/or} =_{def} \langle T', P, r^{(P)} \rangle$ is an AND/OR tree with respect to the property P. If $T_{or} = \emptyset$, the tree $T^{(P)}_{and/or}$ becomes $T^{(P)}_{and}$ AND tree with respect to the property P, if $T_{and} = \emptyset$, the tree $T^{(P)}_{and/or}$ becomes $T^{(P)}_{or}$ OR tree with respect to the property P.

Using the relation $r^{(P)}$, the satisfiability (or deniability) of the property P is propagated across the whole $T_{and/or}$ tree. We say that in $r^{(P)}$ $(x, y_1, \dots, y_n)$ the nodes $y_1, \dots, y_n$ are source nodes and the node x is a *target node*. In other words, the relation $r^{(P)}$ is directional, directed from the source nodes to the target node.

Let as fuzzify the tree $T_{and/or}$ fuzzifying the set of nodes T and the set of edges E: $\tilde{T} = \{x, \mu_T(x) \mid x \in T, \mu_T: T \to [0,1]\}$, where μ_T is a membership function; $\tilde{E} = \{(x,y), \mu_E(x,y) \mid (x,y) \in E, \mu_E: E \to [0,1]\}$, where μ_E is the strength of edge (x,y).

Let us further fuzzify the property P assuming that the strength of the property $\tilde{P}$ at the node t is identical with the value of the membership function at this node:

$$\tilde{P} = \{P(t), \tilde{\mu}_P(t) \mid t \in T, P(t) \in P_{tr}, \tilde{\mu}_P(t) \equiv \mu_T(t)\}. \quad (9)$$

Defining the fuzzy implication relation $\tilde{r}^{(P)}$ $(t, t_1, \dots, t_n)$ between source nodes $t, t_1, \dots, t_n$ and the target node t with respect to the property $\tilde{P}$ on the basis of *generalized modus ponens rule to fuzzy logic* or other fuzzy reasoning approach, the relation $\tilde{r}^{(P)}$ at each node of the fuzzified $T_{and/or}$ infers the value of the property$\tilde{P}$, in the forward manner, starting from the given value of $\tilde{P}$ in the leaf nodes of this tree. In a similar way, the relations can also be defined for backward and even bidirectional inferences.

Definition 53 Fuzzy AND/OR tree

$\tilde{T}^{(\tilde{P})}_{and/or} =_{def} \langle \tilde{T}', P', \tilde{r}^{(P)} \rangle$ is a fuzzy AND/OR tree with respect to the property $\tilde{P}$. If $T_{or} = \emptyset$, the tree $\tilde{T}^{(\tilde{P})}_{and/or}$ becomes a fuzzy $\tilde{T}^{(\tilde{P})}_{and}$ AND tree with respect to the property $\tilde{P}$, if $T_{and} = \emptyset$, the tree $\tilde{T}^{(\tilde{P})}_{and/or}$ becomes a fuzzy $\tilde{T}^{(\tilde{P})}_{or}$ OR tree with respect to the property $\tilde{P}$.

Let now define the terms view, viewpoint and perspective. Definitions of these terms are based on the goal-oriented methodology and view reconciliation methodology, which in the field of Computer Science was originated by SADT methodology [228] and by Leite PhD thesis [229], and was further developed by many other authors, mainly in software requirements engineering. In line with this methodology and the hypothesis H1, we will define the terms *perspective*, *viewpoint* and *view*. These terms are defined using terms quality attribute (see Definition 2) and linguistic vector (see Definition 37).

In the goal-oriented methodology, *quality requirements* are referred to as *quality goals*. Quality goals define requirements for quality attributes, shortly referred to as *qualities*. The meaning of qualities cannot be defined by formally defined properties, because they are vague concepts. For example, the meaning of '*reliable*' depends on a particular viewpoint and the meaning of '*highly*' cannot be defined precisely at all. The concepts a viewpoint and a perspective have precisely definable meaning, but they are evaluated using uncertain information. Thus, we consider the QoS_{EBS} planning problem as a fuzzy problem. Let $X = \{\chi_{1,\dots,}\chi_n\}$ is a set of EBS qualities, $X_1 \subseteq X$ is a set of generic qualities, and $Y_i \subset X$ is a set of bottom-level subqualities of the quality $\chi_i \in X_1$

Now for each quality $\tilde{\chi} \in \{X_1 \cup Y_1 \cup \ldots \cup Y_m\}$, where m is a number of generic qualities we define a linguistic variable $Q_{\tilde{\chi}}$. All defined linguistic variables share the set of linguistic labels

$$L_{tr} = (unsatisfied, \ldots, satisfied), \quad (10)$$

which names their linguistic terms. For each linguistic term *trm* of each UoD of the quality $\tilde{\chi}$ we define its value (i.e. a fuzzy variable)

$$\Gamma_{\tilde{\chi}} = \{(x_{ij}, \mu_{ij}) \mid x_{ij} \in dom_{trm}(\chi_i), \mu_{ij}(x_{ij}): dom_{trm}(\chi_i) \to [0,1]\}, \quad (11)$$

where dom_{trm} is a part of the UoD of the quality $\tilde{\chi}$ associated with the linguistic term *trm*.

For a particular EBS, the developers are free to choose any names of linguistic values. In other words, the number of qualities, number of linguistic terms and their labels depends on a particular EBS. For example, the labels *below low quality* (synonym to *unsatisfied*), *low quality*, *average quality*, *high quality*, *perfect quality* (synonym to *satisfied*) can be defined.

Qualities form hierarchies (or taxonomies), which are modelled by the fuzzy AND trees (see Definition 52). Top level qualities are referred to as *generic qualities.* For a particular EBS in questions developers define own list of qualities.

Definition 54 Perspective

A perspective is the role-dependent angle under which one sees EBS and takes a viewpoint-based judgement on the acceptable quality level of this service.

Such judgement is always biased by the role-related understanding what midpoints in the part between service and its requestor are most important.

In line with Fig. 4 discussed in Section 3.1, we provide following perspectives:

- $\boldsymbol{\pi_1}$ **is the presentation perspective**. This perspective is related to service requestors, which usually focuses on the quality of the presentation of information produced by this EBS for service requestors. To some extent this overlaps with the data quality mentioned in [1]. For service requestor most important are such qualities as the relevance, granularity, and level of detail of presented information; its accuracy, consistency, completeness, and timeliness; appropriateness of its visualisation, perspicuity, and transparency for a service requestor; and etc. One of problems arising in the context of SoES in evaluating QoS_{EBS} from this perspective is the separation of concerns between EBS itself and software, which implements a service consumer. The latter impacts rather the quality of the whole SoES than the quality of a particular EBS because it is used by service requestor for all required EBSs.

- $\boldsymbol{\pi_2}$ **is the transportation perspective**. This perspective is related to computer network administrators, which usually focus on such qualities as the response time, maximal throughput, service availability, networks reliability, etc. In SoES context, the problem of the separation of system and service concerns arises again.
- $\boldsymbol{\pi_3}$ **is the infrastructure perspective.** This perspective is related to SoES platform administrators, which usually focus on such qualities as performance, reliability, security, and etc. In SoES the implementation platform usually is shared among many or even all EBSs. So, the problem of the separation of system and service concerns arises again.
- $\boldsymbol{\pi_4}$ **is the web service perspective.** This perspective is related to web service developers which along with the reliability, security and other WS as a product related issues usually focus on such WS as a service qualities as messaging, responsiveness, courtesy (politeness, respect for service requestor, friendliness, etc.).
- $\boldsymbol{\pi_5}$ **is the application perspective.** This perspective is related to applications developers. As a rule, the functionality of an EBS is implemented by some application, i.e. by some software component. Depending on the implementation platform, the components are implemented differently, for example in Microsoft Windows Communication Foundation [133] they are implemented as *service*

classes. The non-functional properties of the application affect the quality of the whole EBS. The application perspective focuses on non-functional properties of application (software product) or, in other words, on EBS technical quality.

- $\boldsymbol{\pi_6}$ **is the data perspective.** This perspective is related to data administrators. According to [230] and many other sources, SoES provides a special kind of services – SoES data services. In such services, web services are used to encapsulate data and the supported behaviour, for example, the operations that manipulate the data. The term data is used here to address data stored in the enterprise's data bases as well as XML documents and various contents:

"...a single data service will usually only expose or manipulate a core set of data, rather than all data for the entire enterprise" [230].

It is obvious that the quality of encapsulated data essentially affects the QoS_{EBS} of the whole EBS as well as quality of components processing these data. Thus, the data perspective focuses on the quality of encapsulated data.

- $\boldsymbol{\pi_7}$ **is the domain perspective.** This perspective is related to business domain experts, which focuses on qualities specific to a particular business domain, for example, for online banking services or for online streaming multimedia services. In addition, even the qualities defined in all business domains, in different domains can be treated differently because of some practical reasons [137]. The specific nature of a particular domain may effects weights assigned to the values of some EBS qualities. For example, in online streaming multimedia services the quality *bits-per-second* is more important than the *security*. In online banking services, vice versa, the security is more important than the *bits-per-second* [231]. *Inter alia*, despite the fact that media applications, including video-oriented ones, also emerge in SoES, up to date they are rather marginal there (an exhaustive discussion on the QoS of video-oriented services can be found in [232]). In SoES, the domain perspective focuses on internal enterprise's domains, for example, on manufacturing or human resource management.

- $\boldsymbol{\pi_8}$ **is the socio-economic perspective.** This perspective is related to business experts who focus usually on business, economic and social issues. The price of service, payment mode (e.g., kinds of accepted bank cards), legal constraints, and other similar issues are regarded as most important EBS qualities. *Business effect of the service* is one of the most important socio-economic characteristics for the EBS. We ignore the fact that, in principle, the socio-economic perspective can be splitted into several finer-grained perspectives.

The above presented list of perspectives is only illustrative. For any particular EBS in question its developers can define own list of perspectives.

Deciding about acceptable QoS_{EBS} quality level, the decider takes into account not only his role-specific attitudes but also his understanding of what quality is in general (i.e. his viewpoint on quality). It means that decisions should be classified according to perspectives inside of each viewpoint. In the context of QoS_{EBS} planning problem, such decisions should be presented in the form of linguistic vector (see Definition 37), each component ow which represents a generic quality defined for the EBS in question.

Definition 55 Viewpoint

A viewpoint is a general understanding what the quality is, on the basis of which one takes a judgement on the acceptable quality level of an EBS. In the context of QoS planning problem, viewpoints integrate perspectives.

According to Sommerville and Sawyer [17] there are two kinds of viewpoints: viewpoints associated with a particular role and viewpoints reflecting a particular role-in depended standpoint. In line with this claim and the hypotheses H1, we define the following viewpoints:

- $\boldsymbol{\omega_1}$**is the metaphysical viewpoint**. According to this viewpoint, a quality of EBS is a degree of excellence where excellence is defined as an abstract ideal, which shows the direction where services are heading to but possible will never get there.

- $\boldsymbol{\omega_2}$ **is the cost-based viewpoint**. According to this viewpoint, a quality of EBS is a degree of excellence at an acceptable price.
- $\boldsymbol{\omega_3}$ **is the value-based viewpoint**. According to this viewpoint, a quality of EBS is service fitness for requestor's values and preferences. It differs depending on a service requestor for whom it is defined.
- $\boldsymbol{\omega_4}$ **is the pragmatic viewpoint**. According to this viewpoint, a quality of EBS is the balance of features and qualities of service that bear on its ability to satisfy stated or implied needs of service requestor. It depends on a particular context, in which the service is consumed or, in other words, the judgment about the quality of a service depends on the aims and goals for which this service is intended to be used.
- $\boldsymbol{\omega_5}$ **is the provider's viewpoint**. According to this viewpoint, a quality of EBS is a compliance with the stated requirements, which are mostly formulated in business and technical terms.
- $\boldsymbol{\omega_6}$ **is the designer's viewpoint**. According to this viewpoint, a quality of EBS is something that is defined by the values of quantifiable and measurable internal properties of a service. This viewpoint assumes that the greater the amount of a desired attribute is possessed by a service, the higher is the quality of this service.

The above presented list of viewpoints is only illustrative. For any particular EBS in question its developers can define own list of viewpoints.

Similar as in case of perspectives, viewpoint-based decisions on the acceptable EBS quality level should be presented in the form of linguistic vector (see Definition 37). For each viewpoint this vector aggregates vectors of perspectives that are associated with this viewpoint.

Definition 56 View

A view is viewpoint- and perspective-independent judgement on the acceptable quality level of EBS in question.

This judgement is a linguistic vector produced by aggregation linguistic vectors of viewpoint.

Definition 57 QoS for EBS

QoS_{EBS} is a linguistic vector produced by view-level judgement on the acceptable quality level of this EBS.

Chapter 5 Development of Problem Fuzzification Methodology

The chapter proposes a methodology to guide problem fuzzification process. **Section 1** describes this methodology. **Section 2** concludes the chapter.

5.1. A Methodology to Guide Problem Fuzzification Process

The proposed problem-independent MF construction methodology is based on the ideas, described in [233,234]. It is presented in Fig. 7. The methodology provides 10 steps, starting from the analysis of the problem in question and finishing by the definition of the MF that is most suitable for this problem. By a problem we mean the construction of MFs taking into account the allowed degree of subjectivity, sources of input data, data collection methods, etc. (see Fig. 6). The methodology provides a number of backtrackings to the previous steps when it is necessary that the obtained results should be refined. In Fig. 7, the backtrackings are shown by dotted lines. Step 10 is required only in cases, where MFs are constructed using automatic approaches. The main scheme of the proposed approach is as follows. First of all, we should decide which property of the object under consideration should be modelled, and which MF construction approach should be chosen. Further, UoD of this property should be categorized, the linguistic variable (including linguistic terms) should be defined, and MF should be constructed, verified, and validated. If an automatic MF construction approach was applied, MF may be improved using the appropriate learning algorithms.

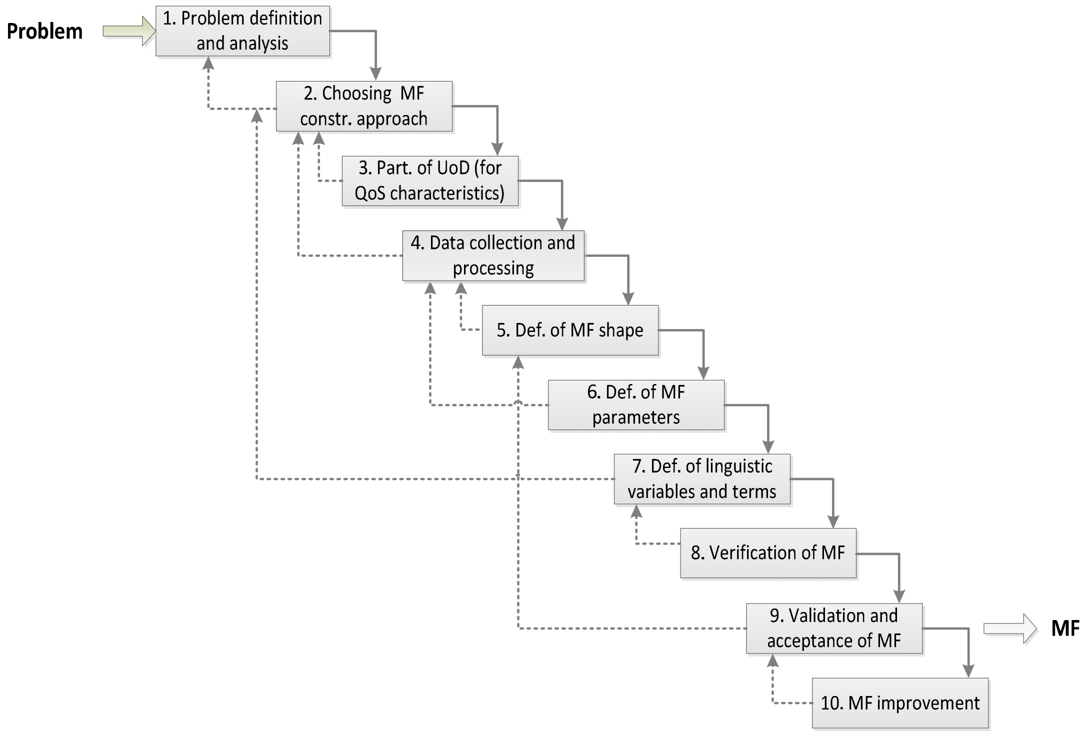


Fig. 7 MF construction methodology

A more detailed description of the steps, shown in Fig. 7, is presented below:

1. The fuzzy modelling problem under consideration should be defined and analysed. It means that it should be decided which properties of the object (or objects) in question should be modelled and UoD should be defined for each of these properties: discrete and finite or continuous and infinite. Further, specification MF requirements should be developed. The specification should define: a) allowed degree of subjectivity of MF; b) allowed problem-related bias; c) the kind of data used to extract knowledge about the shape and parameters of MFs; d) necessity to justify these results; e) automatic MF construction approach (if applicable); f) how – directly or indirectly – subjective experts' evaluations should be reflected (if applicable); and g) kind of questionnaires. Some problem-specific requirements may be added.
2. On the basis of requirements specification, a MF construction approach (a branch in Figure 2) should be chosen for each property.

3. UoDs of each property should be partitioned into categories according to the chosen criteria (e.g. categories of the quality or temperature). To unify the understanding of linguistic terms defined for different linguistic variables (in our case, for different quality attributes) the most suitable conversion scale for transforming linguistic terms into fuzzy numbers (e.g. scales proposed in [235] or in [236]) should be chosen and further used by all experts.
4. The data required for extracting knowledge about shape and parameters of MFs should be collected and processed.
5. On the basis of the obtained results, the shape of MF (e.g. triangular, trapezoidal, L-shaped, Gama-shaped, Sigmoidal, etc.) is determined.
6. The parameters of MF are defined. The number and meaning of the parameters depend on the shape of a function. For example, triangular MF is defined by 3 parameters that define the three corners of the underlying triangular, and Gaussian MF is defined by 2 parameters that define the centre and width of this function graphic.
7. A linguistic variable (including linguistic terms) should be defined for each property under consideration.
8. Verification of MF is performed. The MF verification is checking whether MF complies with its requirements specification.
9. Validation of MF is a process of making sure that the MF really captures the intended meaning of the linguistic terms in the best way.
10. Improvement of MF is usually performed by learning. The improvement is going in a cycle until, finally, MF is accepted. Artificial neural networks, genetic algorithms, and other machine learning methods can be used for this aim.

The presented Example 3 analysis has demonstrated the applicability of this proposed methodology in the examined context of QoS_{EBS}. On the other hand, this example has shown that the construction of MFs is far from being a simple task and the degree of subjectivity and problem bias fully depends on the experts' selection procedures.

Example 3 Construction of membership functions for QoS_{EBS}

Let us present an illustrative example how to apply the proposed methodology in construction of MFs for QoS_{EBS} attributes. The steps of applying the MF construction methodology, using the performance attribute, are explained as follows:

1. *Problem definition and analysis.* Context description: In a Service-oriented Enterprise System, a new EBS, namely, an Invoice Submission service, should be developed. The quality of this service should be preliminary planned or, in other words, the property '*quality*' of the object '*Invoice Submission service*' should be modelled. We refer to this property as QoS_{EBS}. Syntactically, QoS_{EBS} can be considered as a composition of its attributes. Each QoS_{EBS} attribute has a hierarchical structure and can be represented as a tree of its lower levels sub-attributes. Semantically, QoS_{EBS} can be understood in a number of different ways called viewpoints [237].

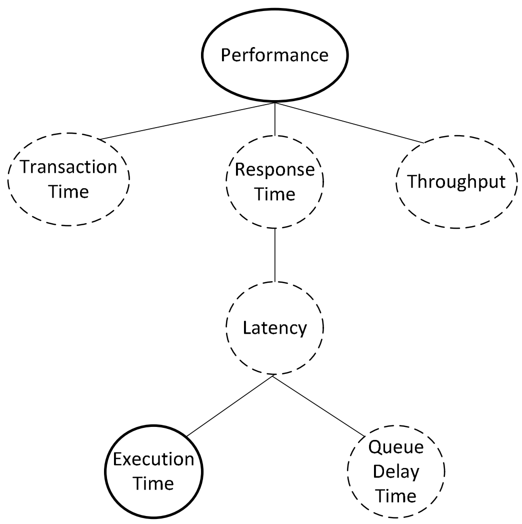


Fig. 8 Decomposition of the performance attribute

Besides, for each viewpoint QoS_{EBS} can be defined from 8 different perspectives: presentation, transportation, infrastructure, web service, application, data, domain, and socio-economic [237]. So, final QoS_{EBS} is defined as a result of aggregation of perspectives and balancing of viewpoints. It is supposed that an expert (or a group of experts), taking into account the specifics of EBS in question, should decide on the common categorization of UoD to the bottom level QoS_{EBS} sub-attributes and on the shape and parameters of MF, which should also be common for all these sub-characteristics. After this, representants of each perspective (they may have different viewpoints on what the quality means) should propose its quality for each bottom-level sub-characteristic plan in terms of common categorization. Finally, the problem is fuzzified and preliminary QoS_{EBS} is calculated using the methods described in [237].

Problem statement: For simplicity, we consider only one bottom-level sub-attribute, namely, *Execution time* of the attribute *Performance* (Fig. 8), i.e. the values of execution time range in the interval $(0,+\infty)$. In Fig. 8, this attribute is placed in a box, outlined by a thick blue line. Its UoD is continuous and infinite. Besides, in this example, we deal only with perspectives and, for the sake of simplicity, ignore different viewpoints on the nature of quality.

MF requirements specification:

a) The *allowed degree of subjectivity*: Subjectivity of MF should be minimised.
b) The *allowed problem-related bias*: The problem-related bias should be minimized. Expert evaluations should take into account the specificity of EBS under consideration. It means that the expert group should include at least one expert familiar with this specificity.
c) *Data requirements*: Empirical data should be used to extract knowledge about the shape and parameters of MFs. Data should be collected applying the phenomenography-based methodology. Relevant sources of literature should also be used.
d) *Justification of results*: The shape and parameters of MF should be justified using MF construction through experiment techniques.
e) *Automatic MF construction approach*: Not applicable.
f) *Reflection of experts' evaluations*: MF should reflect expert evaluations directly.
g) *Kind of questionnaires*: MF exemplication.
h) *Problem-specific requirements*: Static MF should be constructed. The shape of MF should also be applicable (with different parameters) to the

fuzzification of sub-attributes Transaction time, Throughput, and Queue delay.

2. *Choosing MF construction approaches*. On the basis of MF requirements specification, described above, the direct MF construction approach with multiple experts was chosen. The selection of MF construction approaches is shown in Fig. 9.

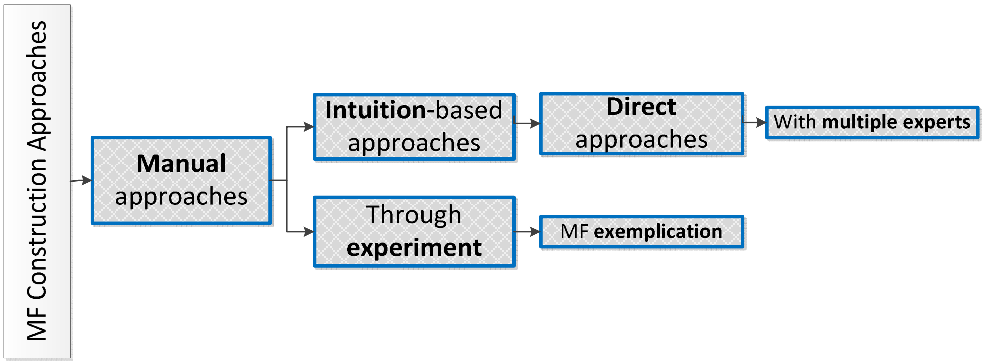


Fig. 9 Selected MF construction approaches

3. *Partitioning of UoD*. UoD of Execution time is partitioned into 3 categories of quality: Low, Moderate, and High (see Fig. 10).

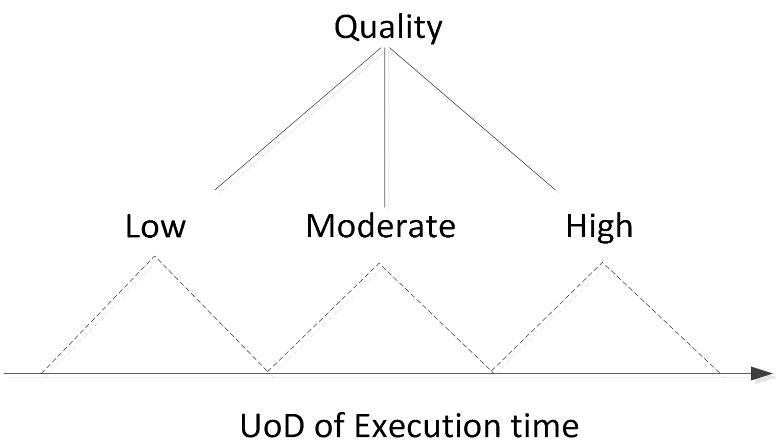


Fig. 10 Partitioning of the Quality

4. *Data collection and processing*. In order to minimize the subjectivity of MF, an intuition-based expert judgement approach was combined with a perspective-based approach. In order to minimize the degree of subjectivity and problem-related bias, a group of 8 experts – 8 representing different perspectives was formed. The experts expressed their opinion on partition intervals of linguistic terms High, Moderate, and Low in UoD of Execution

time, and the shape of MF. The experts took into account the MFs shape of other sub-attributes of the performance attribute. The collected data are presented in Table 5.

Table 5 Partition intervals of *Execution time*

Exp.	Perspectives	Execution time (*min*)		
		High/Shape	Moderate/Shape	Low/Shape
E1	Presentation	(0.08,0.32)/L-shaped	(0.22,1.61)/triang.	(1.60,2.40)/Γ-shaped
E2	Transportation	(0.01,0.31)/L-shaped	(0.15,1.57)/triang.	(1.45,2.10)/Γ-shaped
E3	Infrastructure	(0.07,0.31)/L-shaped	(0.38,1.58)/triang.	(1.37,1.90)/Γ-shaped
E4	Web Service	(0.08,0.28)/L-shaped	(0.22,1.38)/triang.	(1.28,1.80)/Γ-shaped
E5	Application	(0.05,0.28)/L-shaped	(0.13,1.41)/triang.	(1.23,1.85)/Γ-shaped
E6	Data	(0.001,0.26)/L-shaped	(0.12,1.27)/triang.	(1.15,1.80)/Γ-shaped
E7	Domain	(0.01,0.24)/L-shaped	(0.25,1.22)/triang.	(1.23,1.90)/Γ-shaped
E8	Socio-economic	(0.05,0.29)/L-shaped	(0.13,1.47)/triang.	(1.37,2.15)/Γ-shaped

After the discussions, the experts agreed on the following ranges of QoS intervals: Low = (1.3,1.99), Moderate= (0.2,1.45), and High = (0.03,0.29).

5. *Definition of the MF shape*. On the basis of step 4, the Gama-shaped MF for *Low* linguistic term, triangular shape of MF for *Moderate* linguistic term, L-shaped MF for *High* linguistic term, and have been chosen.

6. *Definition of MF parameters*. The MF parameters are as follows: *Low* = (1.3,1.99) (Gama-shaped, defined by 2 parameters); *Moderate* = (0.2,0.83,1.45) (triangular MF, defined by 3 parameters), and *High* = (0.03,0.29) (L-shaped MF, defined by 2 parameters).

7. *Definition of linguistic variables and terms*. The linguistic variable Quality is defined as follows:

$$Quality = \left\langle \begin{array}{l} Execution\,Time,\ \{Low, Moderate, High\}, (0,+\infty),\quad M(Low) = \begin{cases} 0 & if \quad x \le 1.3 \\ \dfrac{x-1.3}{0.69} & if \quad 1.3 < x \le 1.99 \\ 1 & if \quad x > 1.99 \end{cases}, \\ M(Moderate) = \begin{cases} 0 & if \quad x \le 0.2 \\ \dfrac{x-0.2}{0.63} & if \quad 0.2 < x \le 0.83 \\ \dfrac{1.45-x}{0.62} & if \; 0.83 < x < 1.45 \\ 0 & if \quad x \ge 1.45 \end{cases},\quad M(High) = \begin{cases} 1 & if \quad x = 0 \\ 1-\dfrac{1}{0.29}x & if \; 0 < x \le 0.3 \\ 0 & if \quad x > 0.29 \end{cases} \end{array} \right\rangle$$

8. *Verification of MF*. MF was checked whether it complies with its requirements specification described in Step 1. The results are presented in Table 6.

Table 6 Verification matrix

Req. No.	MF requirements	Degree of verification	Verification method
a	Degree of subjectivity	Partly (result of best efforts)	Experts evaluation
b	Problem-related bias	Partly (result of best efforts)	Experts evaluation
c	Data requirements	+	Inspection
d	Justification of results	+	Inspection
f	Reflection of experts' evaluations	+	Inspection
g	Kind of questionnaires	+	Inspection
h	Problem-specific requirements	+	Inspection

9. *Validation and acceptance of MF*. On the basis of MF exemplication experiment, the shapes of MF for linguistic terms High, Moderate, and Low was slightly modified. The final MF parameters are as follows: Low = (1.4,2.0); Moderate = (0.2,0.7,1.5), and High = (0,0.3).

■

5.2. Conclusions

This chapter generalises the ideas of various authors analysed in Section 3.3 and proposes a problem-independent ten step methodology that could be applicable to any particular problem for constructing MF under the assumption that the fuzziness is defined as vagueness. The application of the proposed methodology is demonstrated by example. The main conclusions of this chapter are as follows:

1. The chapter proved the hypotheses H5 because the methodology that guides the fuzzification procedure for any QoS_{EBS} quality attribute and any EBS consideration perspective was developed.
2. From the analysis of presented example follows that the full objectivisation of the fuzzification process is impossible. The human factor still plays significant role in this process and the degree of subjectivity and fuzzification bias significantly depends on the experts' selection procedures.

Chapter 6 Modelling and Planning of Enterprise Business Service Quality

This chapter presents main theoretical results of the doctoral research. **Section 1** formalises the QoS_{EBS} planning problem. **Section 2** builds problem-oriented QoS_{EBS} model. **Section 3** designs an ensemble of collaborating algorithms to solve the QoS_{EBS} planning problem. **Section 4** describes the proposed algorithms. Section 5 describes the architecture and other implementation issues of the proposed QoS_{EBS} planning system.

6.1. Problem Formalisation

Let

- $X = \{\chi_{1,\ldots,}\chi_n\}$[8] is a set of qualities, which models high-level business-oriented EBS quality requirements (e.g., "*a service under consideration should be highly reliable*") and $X_1 \subseteq X$ is a set of generic qualities;
- $\Omega = \{\omega_i \mid 1 \le i \le 6\}$ is a set of viewpoints and $\vec{\Omega} = \{\vec{\omega_i} \mid 1 \le i \le 6\}$ is a set of weighted linguistic vectors associated with these viewpoints;
- $\Pi = \{\pi_i \mid 1 \le i \le 8\}$ is a set of perspectives and $\vec{\Pi} = \{\vec{\pi_i} \mid 1 \le i \le 8\}$ is a set of weighted linguistic vectors associated with these perspectives;
- $Q = \{Q_1, \ldots, Q_m\}$ is a set of linguistic variables associated with the generic qualities and $Q' = \{Q'_1, \ldots, Q'_p\}$ bottom-level subqualities of generic qualities;
- $\rho_{eqlb}^{Q'}$ is a labelled equilibrium fuzzy relation on Q' (see formula 16);
- ρ^{Ψ} is a fuzzy relation relating viewpoints, perspectives and linguistic variables associated with generic qualities (see formula 13);
- $\rho^{\Psi_k}, 1 \le k \le 6$ is a family of relations produced by projection of ρ^{Ψ} to $\Pi \times X_1$ (see formula 13);
- $\Phi = \left\{\tilde{T}_{and}^{\tilde{\psi}_{i,j}^k} \middle| \ 1 \le k \le 6, 1 \le i \le 8,\ 1 \le j \le n,\ n \le N_\Gamma\right\}$, is a family of fuzzy AND trees of EBS quality attribute, where each tree describes a

[8] In this section we use notation defined on the page 85.

generic quality $\chi \in \mathrm{X}_1$ evaluated from a perspective $\pi_j \in \Pi$ and observed from a viewpoint $\omega_k \in \Omega$.

Then the $\mathrm{QoS}_{\mathrm{EBS}}$ planning problem formally is defined by a tuple

$$\langle\, \Omega, \vec{\Omega}, \Pi, \vec{\Pi}, \mathrm{X}, \mathrm{X}_1, \mathrm{Q}, \rho^{\Psi}, \mathrm{X}_2, \mathrm{Q}', \rho^{\mathrm{Q}'}_{\mathrm{eqlb}}, \Phi, \mathrm{Input}, \mathrm{Output}\rangle, \tag{12}$$

where

- **Input** is a set of initial linguistic values of variables from Q or, in other words, linguistic terms assigned for bottom-level qualities of EBS in question evaluating their acceptable quality levels for each perspective observed from the each viewpoint;
- **Output** is a final linguistic vector describing $\mathrm{QoS}_{\mathrm{EBS}}$ (see Definition 57.

This problem is defined on the $\mathrm{QoS}_{\mathrm{EBS}}$ quality model defined in the next section.

6.2. Modelling of $\mathrm{QoS}_{\mathrm{EBS}}$

Let we define a relation

$$\rho^{\Psi} = \{(\psi, \mu_{\Psi}) \mid \mu_{\Psi} :\ \Psi \to [0,1], \psi \in \Psi, \Psi \subseteq \Omega \times \Pi \times \mathrm{Q}\}, \tag{13}$$

which relates viewpoints, perspectives and linguistic variables associated with the generic qualities qualities. For each fixed viewpoint $\omega_k \in \Omega, 1 \le \mathrm{k} \le 6$, the projection of ρ^{Ψ} to $\Pi \times \mathrm{Q}$ produces a family of relations

$$\rho^{\Psi_k} = \left\{ \begin{matrix} \left(\psi^k_{i,j}, \mu_{\Psi_k}\left(\psi^k_{i,j}\right)\right) \Big| \mu_{\Psi_k} : \Psi_k \to [0,1], \psi^k_{i,j} = \left(\pi_i, Q_j\right), \pi_i \in \Pi, \\ Q_j \in \mathrm{Q}, 1 \le k \le 6, 1 \le i \le 8, 1 \le j \le m, m \le N_{\mathrm{Q}} \end{matrix} \right\} \tag{14}$$

which relates perspectives observed from the viewpoint $\omega_k$and linguisic variables for generic qualities on which this viewpoint focuses. In other words, for each viewpoint we have a matrix of linguistic variables

$$\Psi_k = \begin{pmatrix} \psi^k_{1,1} & \cdots & \psi^k_{1,n} \\ \vdots & \ddots & \vdots \\ \psi^k_{8,1} & \cdots & \psi^k_{8,n} \end{pmatrix}. \tag{15}$$

Let we define further a labelled equilibrium fuzzy relation $\rho^{Q'}_{eqlb}$on Q’

$$\rho_{eqlb}^{Q'} = \left\{ \left((Q'_1, Q'_2), \mu_{\rho_{eqlb}^{Q'}}(Q'_1, Q'_2), label \right) \middle| \mu_{\rho_{eqlb}^{Q'}} : Q' \to [0,1], \right.$$

$$(Q'_1, Q'_2) \in Q' \times Q', value(Q'_1) + value(Q'_2) \le C_{eqlb}^{(Q'_1,Q'_2)} \le 1, label \in$$

$$\left. \{\ll, \gg, \sim, \ll\sim, \sim\gg\} \right\}, \tag{16}$$

where value (Q') is a normalized fuzzy number that stands for the given linguistic value of the linguistic variable Q'. Normalisation means that fuzzy numbers are mapped to the interval [0,1]. More exactly, fuzzy numbers for Q'_1 and Q'_2 are calculated according the following formulas:

value (Q'_1)= [Q'_1] ⊘ [Q'_2]; (17)

value (Q'_2)= [Q'_2] ⊘ [Q'_1] - 1. (18)

[ ] is the operator that converts a linguistic value to its fuzzy number.

$C_{eqlb}^{(Q'_1,Q'_2)}$ is an equilibrium constant, which means that a sum *value*(Q'_1)⊕ *value*(Q'_2) cannot exceed this constant, which, in turn, cannot exceed the 1. Or, in terms of fuzzy intervals, the sum of corresponding subintervals cannot exceed the interval defined by this constant which, in turn, cannot exceed the length of interval [0,1].

The label of this relation tells how, if it is necessary, the lengths of subintervals $\mu_\Gamma(x_1)$ and $\mu_\Gamma(x_2)$ should be changed in order to preserve the equilibrium defined by C_{eqlb}: (1) $\ll$ means that the length of subinterval $\mu_\Gamma(x_1)$ should be changed; (2) $\gg$ means that the length of subinterval $\mu_\Gamma(x_2)$ should be changed; (3) $\sim$ means that the lengths of both subintervals should be changed proportionally; (4) $\ll\sim$ means that the lengths of both subintervals should be changed taking preference to $\mu_\Gamma(x_2)$; and (5) $\sim\gg$ means that the lengths of both subintervals should be changed taking preference to $\mu_\Gamma(x_1)$. The number of *EBS quality characteristics* under consideration and their nature depend on a particular EBS under consideration.

Let $X' = \langle X, <_X, F'' \rangle$ is a ranked set on X. For each element $\psi_{i,j}^k$ of matrix Ψ_k we define a fuzzy AND tree

$$\tilde{T}_{and}^{\left(\tilde{\psi}_{i,j}^{k}\right)} = \langle \tilde{X}', \tilde{\psi}_{i,j}^{k}, \tilde{r}^{(\psi_{i,j})} \rangle, \tag{19}$$

where $1 \leq k \leq 6,\ 1 \leq i \leq 8,\ 1 \leq j \leq n,\ n \leq N_{\Gamma}$ of EBS quality characteristics. The value of a linguistic variable $\tilde{\psi}_{i,j}^{k}$ describes a γ_j aspect of EBS (e.g., reliability or security) from the perspective π_i of the viewpoint ω_k. The equilibrium relation $\rho_{eqlb}^{Q'}$ can relate any two leaf nodes of the same or different trees. Using the relations $\tilde{r}^{(\psi_{i,j})}$ and $\rho_{eqlb}^{Q'}$, the given linguistic values can be propagated from leaf nodes up to the root of the tree $\tilde{T}_{and}^{\left(\tilde{\psi}_{i,j}^{k}\right)}$.

The columns of the matrix (15) are vectors, which for each quality describe its evaluations from each perspective defined for viewpoint ω_k. These vectors should be aggregated in order to obtain a vector of viewpoint ω_k. In order to obtain the vector, which describes the final result (i.e. $\mathrm{QoS}_{\mathrm{EBS}}$), the vectors of all viewpoints should be aggregated.

6.3. Design of an Ensemble of Collaborating Algorithms

In line to the above described $\mathrm{QoS}_{\mathrm{EBS}}$ quality model, the ensemble of collaborative algorithms for solving $\mathrm{QoS}_{\mathrm{EBS}}$ planning problem is proposed in Fig. 11.

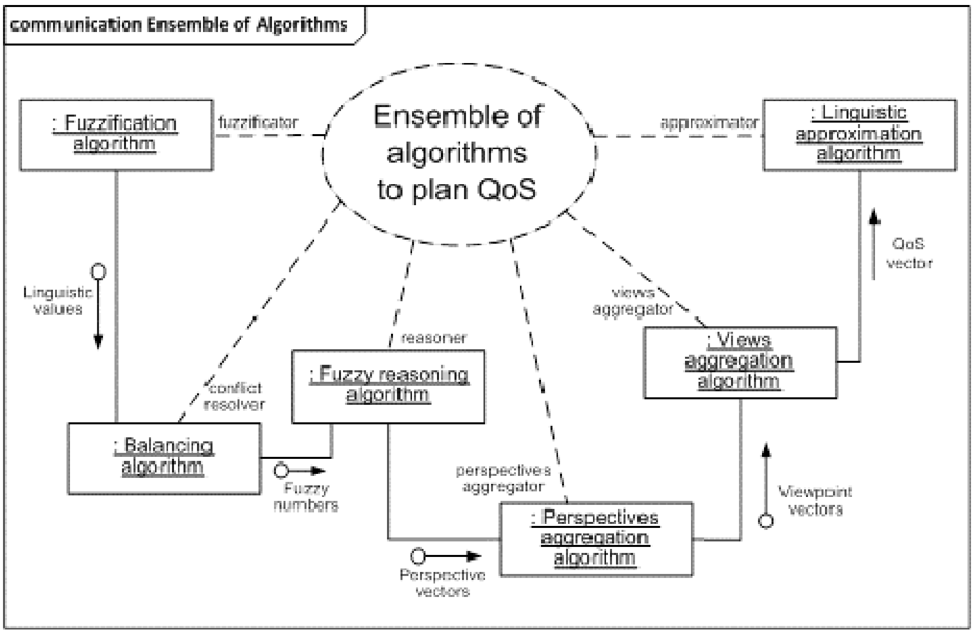


Fig. 11 The ensemble of collaborating algorithms to solve $\mathrm{QoS}_{\mathrm{EBS}}$ planning problem

The ensemble of collaborative algorithms consists of the fuzzification, balancing, fuzzy reasoning, linguistic approximation, and fuzzy aggregation algorithms. The fuzzification algorithm is executed manually by the stakeholders (i.e. service owner, computer network and infrastructure administrators, etc.) and used to fuzzify input data that should be presented for the balancing algorithm in the form of linguistic values of the bottom level quality attributes. Other algorithms should be executed by the QoS_{EBS} planning system.

The cooperation of the algorithms proceeds in the following way:

- for each viewpoint to set linguistic values of input variables and, using fuzzy relation $\rho_{eqlb}^{Q'}$, to resolve conflicts among these values;
- for each fuzzy tree from the family Φ to propagate the defined values forward up to the root of this tree;
- for each viewpoint applying fuzzy graphs union operation to unite trees which describe qualities from different perspectives;
- to infer value of each linguistic variable $\omega \in \Omega$;
- to do linguistic approximation of fuzzy values of all output variables.

All algorithms, except fuzzification and balancing algorithms, are described in the next section. The fuzzification is guided by the methodology proposed in the section 5.1. The balancing algorithm is not described in this dissertation because it is out of scope of the dissertation. This algorithm is a subject of separate research.

6.4. Development and Adaptation of Algorithms

6.4.1. Fuzzy Reasoning Algorithm

According to level order traversal [238], the idea of fuzzy reasoning algorithm is described as follows:

for each viewpoint, for each perspective in this viewpoint, and for each tree of the quality attributes in this perspective perform:

1. Take the linguistic term, which correspond to the fuzzy number, and is assigned to the quality attribute, represented by node, that is located in bottom of left subtree;
2. Using Larsen implication [239], compute the component value (fuzzy number) of higher-level attribute vector, corresponding to in Step 1 mentioned node;
3. Compute by analogy the other components of this attribute vector;
4. Using fuzzy arithmetic mean [240], aggregate all computed components of the attribute vector. Assign the aggregated value to with this vector related quality attribute.
5. Using level order traversal algorithm [238], compute all other to this tree related values of quality attributes.

After traversing of the quality attributes of all trees, related to one perspective, from the obtained values of the top level hierarchy the linguistic vector of this perspective is formed. After the performing reasoning other all perspectives, the weighted average calculation formula [241] to the obtained vectors is used and the vectors of the viewpoints are achieved.

The fuzzy reasoning algorithm is presented below:

Algorithm 1: Fuzzy Reasoning algorithm

```
procedure FuzzyReasoning(viewpoints)
// The result of the procedure is an array of viewpoints vectors.
{
    for each viewpoint in viewpoints
    {
        perspVector = { }  // a vector of perspectives vectors of the viewpoint
        for each perspective in viewpoint.perspectives
        {
            qualities = { }// a vector containing top-level qualities values of the perspective
            for each qtree in perspective.qtrees// iterate over quality hierarchies
            {
            value = CalculateQuality(qtree.Root) // the quality value is a fuzzy number
            Append(qualities, <qtree.qualityName, value>)
             // qualities is a vector of pairs <quality name, quality value>
            }
            Append(perspVector, qualities)
        }
        viewpoint.Vector = AggregatePerspectives(perspVector, viewpoint.perspectiveWeights)
            // AggregatePerspectives(P, W) – a function implementing the aggregation algorithm
            //   (see below)
    }
}
```

```
function CalculateQuality(root)
{
    result = 0
    if (root is leaf)
        result = root.Value
    else
    {
        sum = 0
        count = 0
        subTreeRoot = root.leftmostChild
        while subTreeRoot not NULL
        {
            sum = sum + CalculateQuality(subTreeRoot) * subtreeRoot.edgeValue
            // the multiplication operation means Larsen implication
            // the sum operation means that two fuzzy numbers are summed component by component
            count = count + 1
            subTreeRoot = subTreeRoot.rightSibling
        }
        result = sum / count
    }
    return result
}
```

6.4.2.Linguistic Approximation Algorithm

Linguistic approximation is the one of inherent problem of fuzzy reasoning. The rising question by the obtained result in linguistic approximation is: how to name by a linguistic term a resulted fuzzy set of the deduction process?

For linguistic approximation aim, the distance measurement between two sets of fuzzy numbers, the Euclidean distance equation (Best Fit Technique), is applied:

$$d = \left(A(u_k)B(u_k)\right) = \left[\frac{1}{D}\sum_{k=1}^{D}\left(A(u_k) - B(u_k)\right)^2\right]^{0.5}, \tag{20}$$

where $A(u_k), B(u_k)$ are fuzzy numbers $u_k$of linguistic variables A and B, D is the number of points which describe the shape of MF (D=3 by triangular MF, D=4 by trapezoidal MF).

The linguistic term is obtained as result. Euclidean distances are used to map the resultant fuzzy interval back to linguistic terms.

According to [5,242,243], the idea of fuzzy reasoning algorithm is described as follows:

1. Take the given fuzzy number and given terms of linguistic variable;
2. Determine, among which fuzzy numbers, corresponding to given values of terms of the linguistic variable, fall this fuzzy number;

3. Calculate Euclidean distance among number, mentioned in Step 1, and neighbours number, obtained in Step 2;
4. Determine which of the obtained distance is smaller. If these distances are equal, interactive procedure is used and the expert is asked to choose one of them;
5. Determine what kind of linguistic term correspond the fuzzy number by which the distance is less;
6. Treat this term as linguistic term for approximated fuzzy number, mentioned in Step 1.

The linguistic approximation algorithm is presented below:

Algorithm 2: Linguistic Approximation algorithm

```
function Approximate(A, T)
    // A = (a[1], a[2], ..., a[s]) – a fuzzy number to be linguistically approximated.
    // s – the size (i.e. the number of components) of fuzzy number A.
    // T = (T[1], …, T[nterms]) – an array of linguistic terms of a linguistic variable.
    // T[i] = (t[i][1], t[i][2], …, t[i][s], l[i]) – fuzzy number (t[i][1], t[i][2], …, t[i][s])
    //          and linguistic label (l[i]) of linguistic term T[i].
    // nterms – the number of linguistic terms of the linguistic variable.
{
candidateTerms = { }  // an array of linguistic terms that are close to fuzzy number A.
n = 0          // the number of the terms that are close to A.
for i = 1 to nterms
{
    if (a[1] ≤ t[i][s] AND a[s] ≥ t[i][1]) // if A intersects with with T[i].
    {
        Append(candidateTerms, T[i]]// add T[i] to array candidateTerms.
        n = n+1
    }
}
closestTerms = { }  // an array of linguistic terms closest to fuzzy number A.
minDist = 0  // the least distance between a linguistic term and A.
for i = 1 to n
{
    b = candidateTerms[i]
    d = sqrt( (1/s) Σ_{k=1}^{s} (a[k] − b[k])^2 )
    if (i = 1 OR d < minDist))
    {
        Empty(closestTerms)
        Append(closestTerms, b)
        minDist = d
        m = 1   // the number of terms that are closest to A.
     }
    else
        if d = minDist
        {
            Append(closestTerms, b)
            m = m+1
        }
```

$$d = \sqrt{\frac{1}{s}\sum_{k=1}^{s}(a[k] - b[k])^2}$$

```
 }
if m = 1
    return closestTerms[1]
else
    return interactExpert(closestTerms)
 }
```

6.4.3.Perspective and Viewpoint Aggregation Algorithms

According to [241], let us describe the aggregation in the following way:

$$\mathrm{Aggreg}(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} \mu_{E_i} x_i \tag{20}$$

where x_i is fuzzy numbers for each aggregated node, μ_E are strenghts of edges of the values x_i to the parent node y.

The weights are normalized so, that:

$$\sum_{i=1}^{n} \mu_{E_i} = 1 \tag{21}$$

According to [244], the idea of perspectives aggregation and viewpoints aggregation algorithm is described as follows:

1. Transform the components of the linguistic vectors into fuzzy numbers;
2. Using weight mean formula (20), find component values of aggregated vector.

The perspectives aggregation algorithm is presented below:

Algorithm 3: Perspectives Aggregation algorithm

```
function AggregatePerspectives(P, W)
    // The result of the function is a viewpoint vector – a vector aggregating all perspectives of the viewpoint.
    // P = (P[1], …, P[n]) – an array of the vectors of perspectives.
    // W = (w[1], w[2], … w[n]) – an array of the weights of perspectives.
    // n – the number of perspectives.
    // P[i] = (p[i][1], p[i][2], …, p[i][si]) – a vector containing top-level qualities values of the perspective.
    // p[i][j] = <qname[i,j], qvalue[i,j]> – a pair consisting of quality name and quality value.
{
    resultVector = { }  // an empty resulting viewpoint vector.
    for each qualityName in qualityNames
    {
        qVector = { }  // an array of quality values from all perspectives.
        qW = { } // an array of perspective weights for this quality.
        for i = 1 to n  // iterate over perspectives.
            if exists j such that qname[i][j] = qualityName
            {
            Append(qVector, qvalue[i][j])
            Append(qW, W[i])
            }
```

```
        if qVector.size > 0
        {
            aggregatedQualityValue = AggregateVectorValues(qVector, qW)
            Append(resultVector, <qualityName, aggregatedQualityValue>)
        }
    }
    return resultVector
}

function AggregateVectorValues(T, W)
    // The result of the function is a weighted average of the array of fuzzy numbers.
    // T = (T[1], …, T[n]) – an array of fuzzy numbers.
    // W = (w[1], w[2], … w[n]) – an array of weights.
{
```

$$\text{result} = \frac{\sum_{i=1}^{n} w[i] \bullet T[i]}{\sum_{i=1}^{n} w[i]}$$

```
        // multiplication of number w by fuzzy number A = (a1, a2, ..., an) means the multiplication
        // of each component of a by w, i.e. w•A = (w•a1, w•a2, ..., w•an)
    return result
}
```

The viewpoints aggregation algorithm is presented below:

Algorithm 4: Viewpoints Aggregation algorithm

```
function AggregateViewpoints (P, W)
    // The result of the function is a vector aggregating vectors of all viewpoints.
    // P = (P[1], …, P[n]) – an array of the vectors of viewpoints.
    // W = (w[1], w[2], … w[n]) – an array of the weights of viewpoints.
    // n – the number of viewpoints.
    // P[i] = (p[i][1], p[i][2], …, p[i][si]) – a vector containing top-level qualities values of the viewpoint.
    // p[i][j] = <qname[i,j], qvalue[i,j]> – a pair consisting of quality name and quality value.
{
    resultVector = { }  // an empty resulting viewpoint vector.
    for each qualityName in qualityNames
    {
        qVector = { }  // an array of values of the quality from all viewpoints.
        qW = { } // an array of viewpoint weights for this quality.
        for i = 1 to n  // iterate over viewpoints.
            if exists j such that qname[i][j] = qualityName
            {
            Append(qVector, qvalue[i][j])
            Append(qW, W[i])
            }
        if qVector.size > 0
        {
            aggregatedQualityValue = AggregateVectorValues(qVector, qW)
            Append(resultVector, <qualityName, aggregatedQualityValue>)
        }
    }
    return resultVector
}

function AggregateVectorValues(T, W)
    // The result of the function is a weighted average of the array of fuzzy numbers.
    // T = (T[1], …, T[n]) – an array of fuzzy numbers.
    // W = (w[1], w[2], … w[n]) – an array of weights.
{
```

result = $\frac{\sum_{i=1}^{n} w[i] \bullet T[i]}{\sum_{i=1}^{n} w[i]}$
// multiplication of number w by fuzzy number A = (a_1, a_2, ..., a_n) means the multiplication
// of each component of a by w, i.e. $w \bullet A = (w \bullet a_1, w \bullet a_2, ..., w \bullet a_n)$
return result
}

6.5. Implementation of the System

6.5.1. System Use Cases

Quality planning system was described, using UML diagrams. System Use case diagram is presented in Fig. 12.

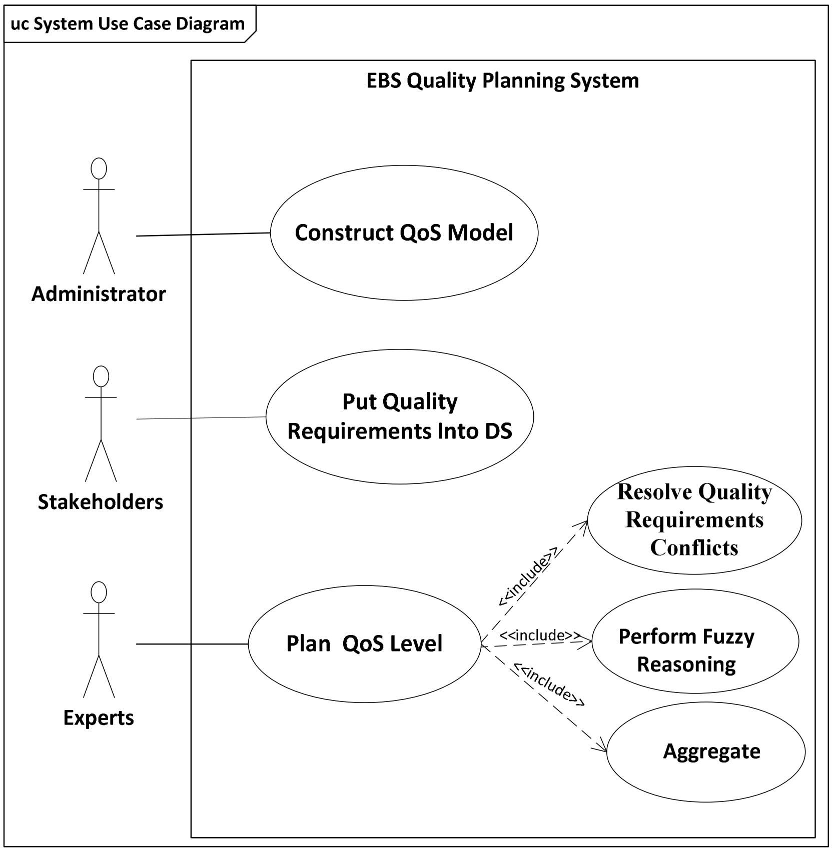


Fig. 12 System use case diagram

The System must perform three use cases:

1. Construct the QoS_{EBS} model;
2. Put data about quality requirements into data storage;
3. Plan QoS level.

The first two use cases are auxiliary, the third – the main. The use case "Plan QoS level" consists of three sub-use cases: "Resolve Quality Requirements Conflicts", "Perform Fuzzy Reasoning", and "Aggregate".

6.5.2.Main Architecture

The following subsystems in the EBS Quality Planning System in order to solve mentioned use cases are distinguished:

The System distinguishes the following subsystems for solving of the above mentioned use cases:

1. The System Administrator Interface subsystem, within which the System receives the data, required for model construction (the list of QoS attributes, the list of perspectives, the list of viewpoints, membership functions and etc.);
2. The User Interface subsystem, within which the stakeholders submit the linguistic values, describing the requirements, which must be met by quality attributes of the bottom hierarchy level. This subsystem is used for input of initial data;
3. Quality Planning Expert's Interface subsystem, within which the System is receiving use cases and the obtained results are returned to the Expert;
4. Model Construction subsystem, which, through System Administrator Interface received data, construct the QoS_{EBS} model;
5. The System Database Forming subsystem that is submitting data from stakeholders into database;
6. Interactive Conflicts Resolving subsystem, which resolves all conflicts, modelled with $\rho_{eqlb}^{Q'}$ relation;
7. Fuzzy Reasoning subsystem is describing the values of the upper hierarchy level quality attributes for each AND a tree of QoS_{EBS} model from fuzzy numbers by conflicts resolving obtained balanced linguistic values, deriving fuzzy numbers;
8. Linguistic Approximation subsystem is approximating in such way obtained fuzzy numbers into corresponding linguistic values and forming for all perspectives with their associated linguistic vectors;
9. Perspective Aggregation subsystem is forming the linguistic vector of this perspective, aggregating with them relating linguistic vectors for

each perspective.

10. Viewpoint Aggregation subsystem is forming the linguistic vector, that describes acceptable quality level of the developed service obtained from aggregating the linguistic vectors of all viewpoints.

The Table 7 shows which subsystems are involved in solving different use cases (referred to use case diagram (see in Fig. 12).

Table 7 Implementation of use cases by subsystems

Use cases / **Subsystems**	Construct QoS Model	Input Quality Requirements Data	Plan QoS Level		
			Resolve Quality Requirements Conflicts	Perform Fuzzy Reasoning	Aggregate
System Administrator Interface	+				
User Interface		+			
Quality Planning Expert's Interface			+	+	+
Model Construction		+			
System Database Forming			+		
Interactive Conflicts Resolving				+	
Fuzzy Reasoning				+	+
Linguistic Approximation					+
Perspective Aggregation					+
Viewpoint Aggregation					+

The decomposition of the system into subsystems is presented in Fig. 13. The Administrator through fill-form interface performs input of the required data, which are important for construction of the model and the input of the values of the bottom level QoSEBS attributes, into Construct Model subsystem. The Expert has the role of conflict resolver and performs this activity through *dialogWindow* interface. The Expert receives the obtained result trough the same interface. The Stakeholders trough *dialogWindow* interface performs input of the linguistic terms. In the inside of the *DataLoading* subsystem, the terms are converted into fuzzy numbers. The QoSEBS class diagram, that describes all classes, related with the construction of model, is presented in Fig. 14. The Class diagram for representation of QoSEBS model in data storage is presented in Fig. 15.

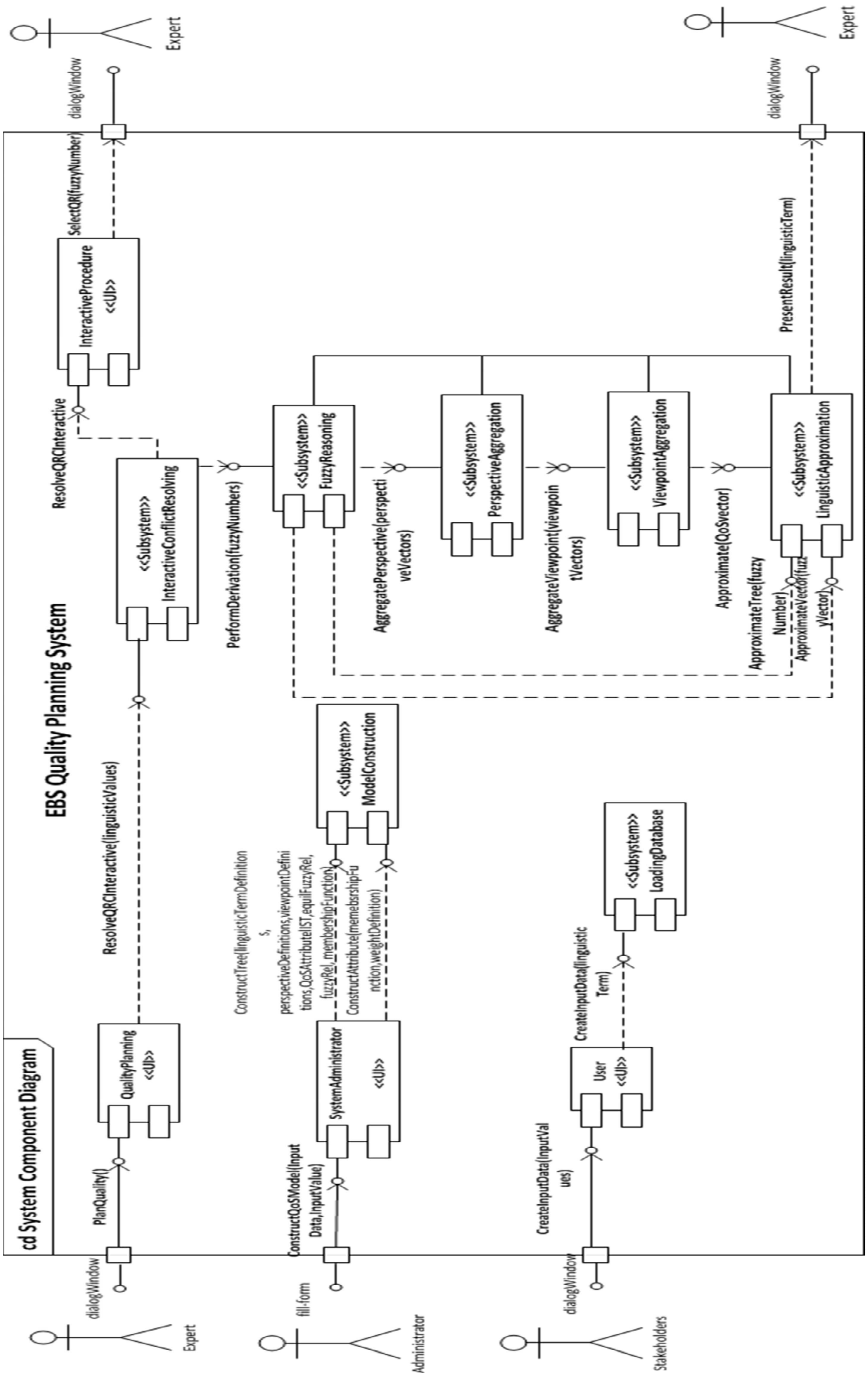


Fig. 13 System Component diagram

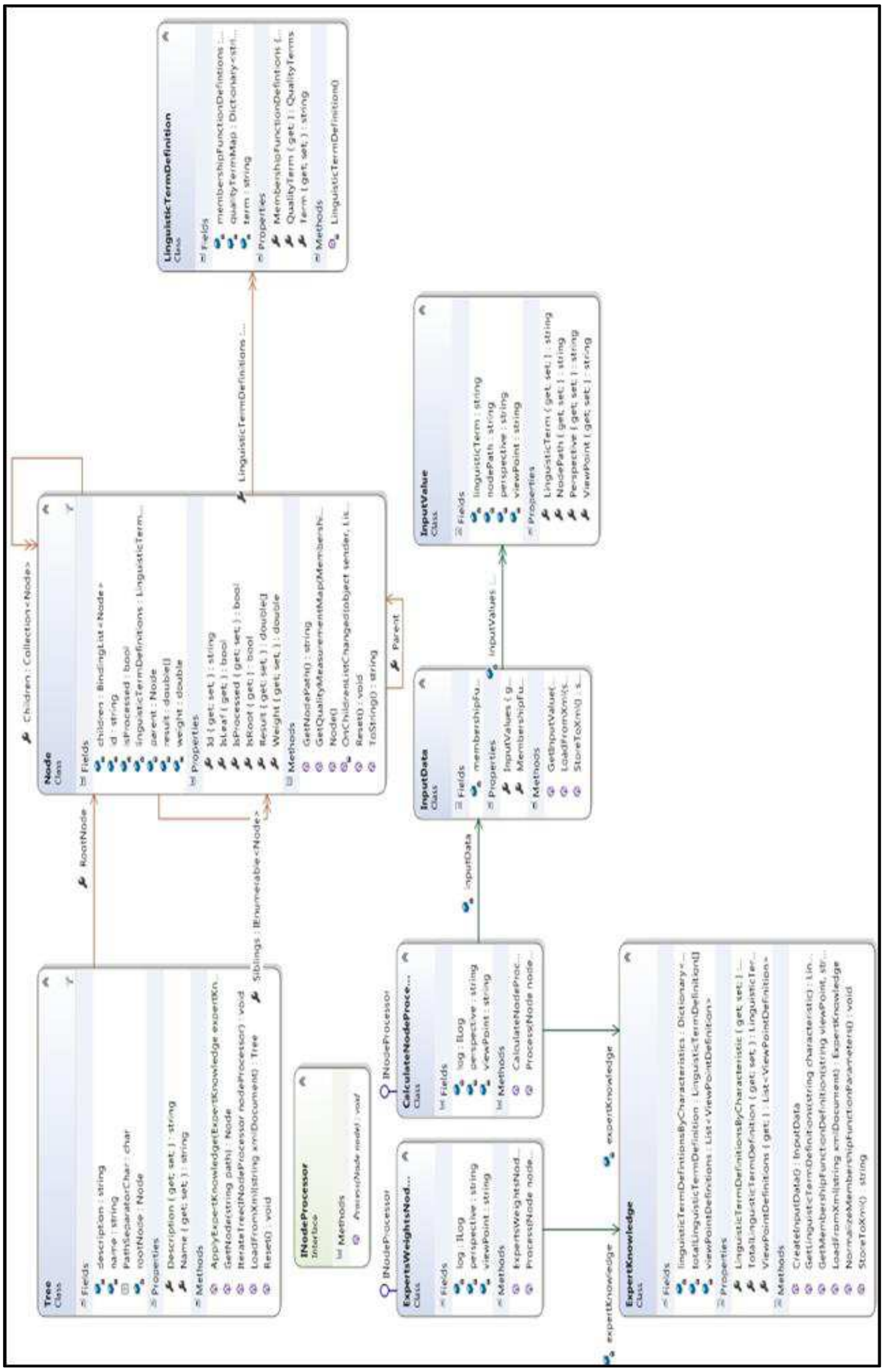


Fig. 14 QoS_{EBS} class diagram

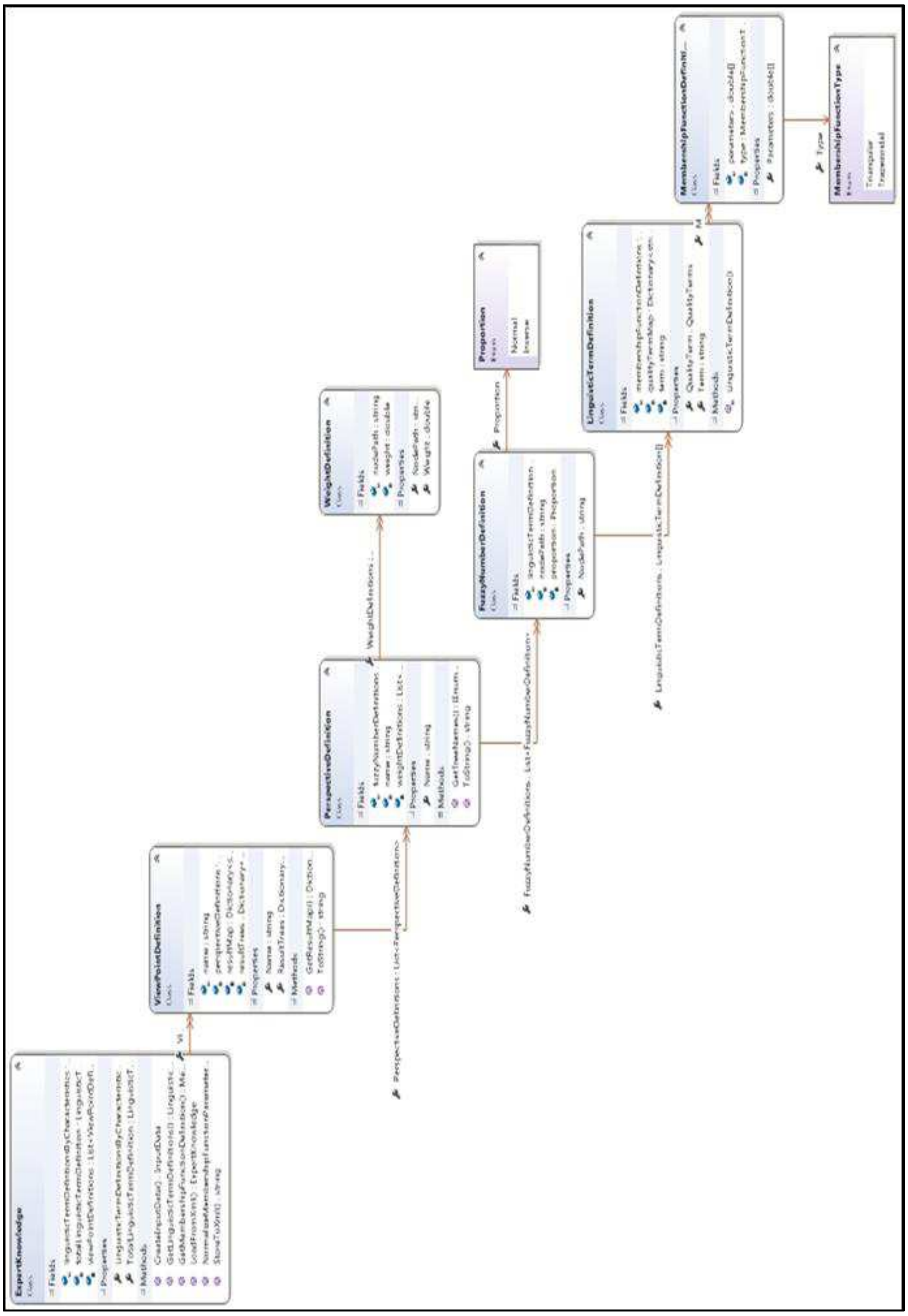


Fig. 15 Class diagram for representation of QoS_{EBS} model in data storage

6.5.3.Implementation of Use Cases *Construct Model* and *Put Quality Requirements Data*

The decomposition of use case *Construct Model* is presented in Fig. 16. The model should be constructed by actor *Administrator*. The sub-use case *Instantiate model* means that class diagram presented in

Fig. 15 should be instantiated by the parameters of a particular EBS (attributes, perspectives, viewpoints, etc.). The sub-use case *Define model* means that data structures to store model data should be created.

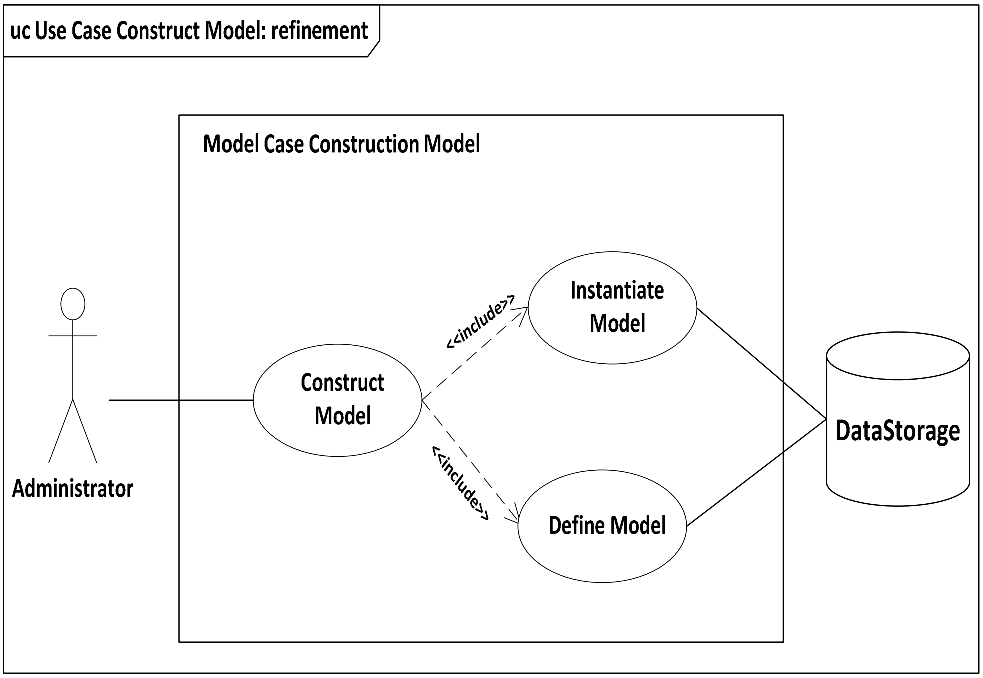


Fig. 16 Refinement of use case *Construct Model*

These sub-uses cases are implemented by the subsystem Model Construction (see Fig. 17). This subsystem includes two components: Instantiate model and Define Model. The interaction between these components and Fill DS with Data subsystem, which implements use case Put Quality Requirements Data, is presented in Fig. 17. This diagram shows also one more component – Data Fuzzification component. This component acts as a pre-processor for Fill DS with Data subsystem. It transforms quality requirements expressed in terms of linguistic values into fuzzy numbers.

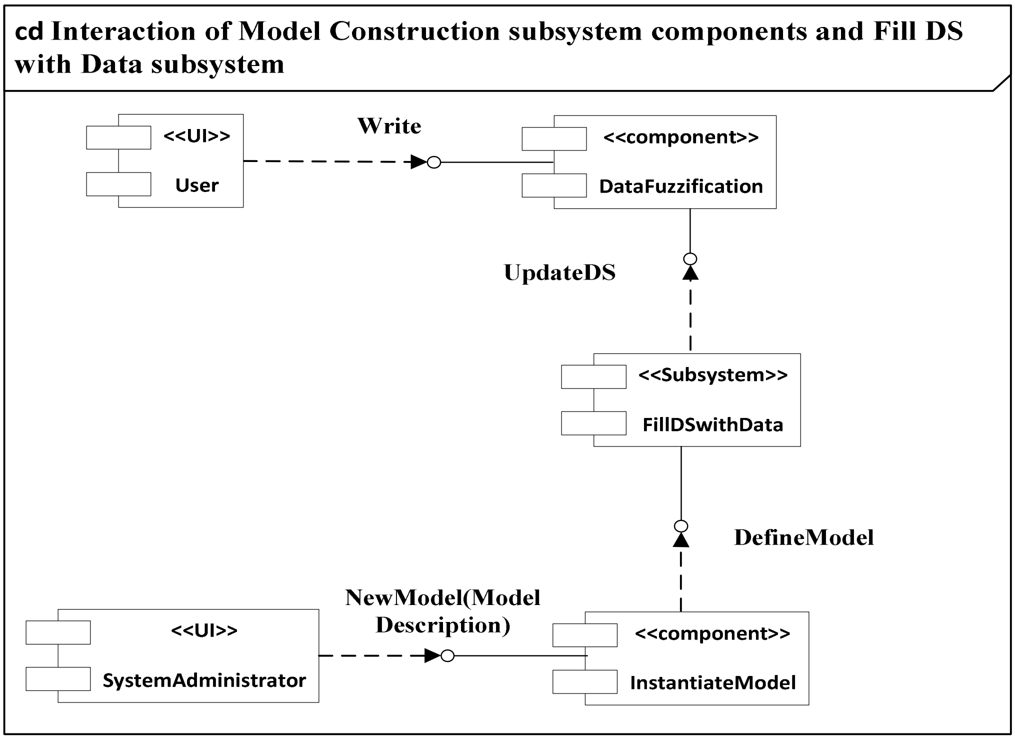


Fig. 17 Interaction of *Model Construction* subsystem components and *Fill DS with Data* subsystem

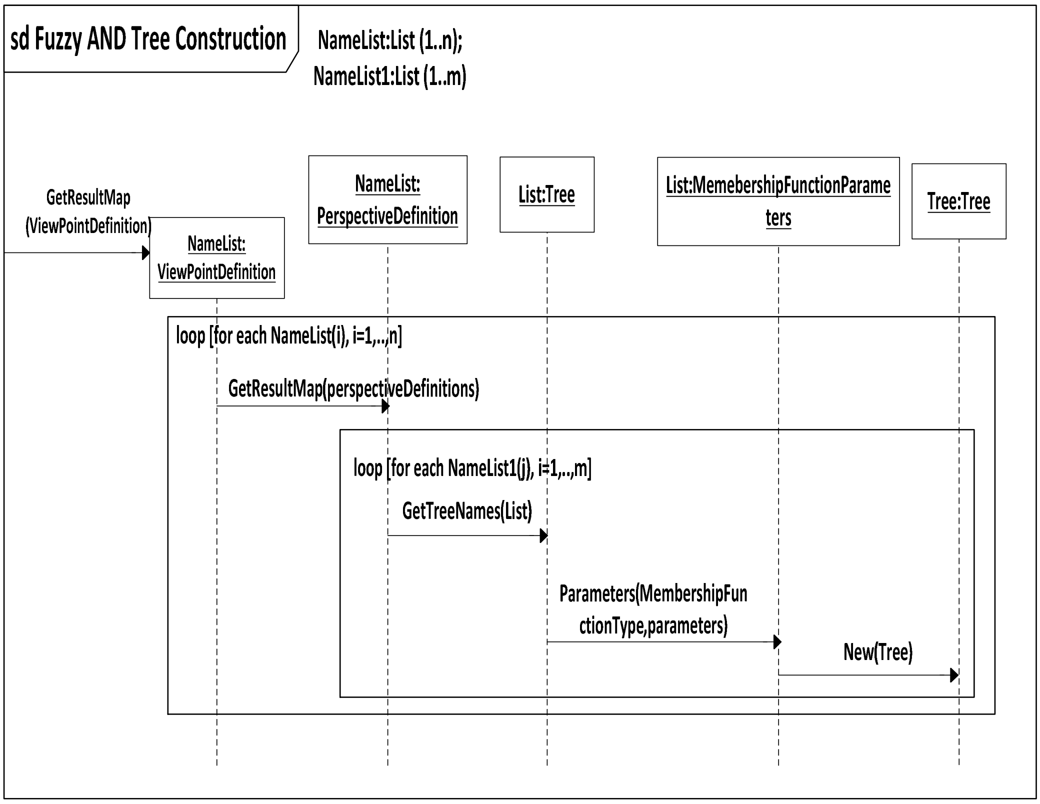


Fig. 18 Sequence diagram for fuzzy AND Tree Construction

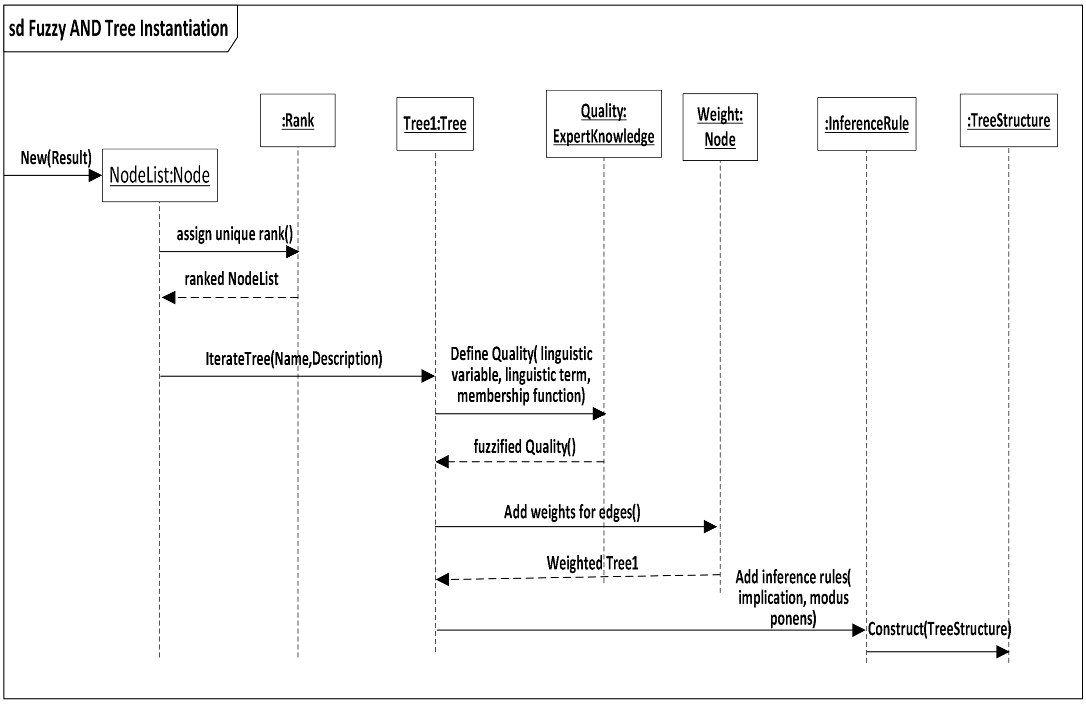


Fig. 19 Sequence diagram for fuzzy AND trees instantiation

Fig. 18 and Fig. 19 illustrate the most important model construction fragment, namely, construction of fuzzy AND trees.

6.5.4.Implementation of Sub-Use Case *Perform Fuzzy Reasoning*

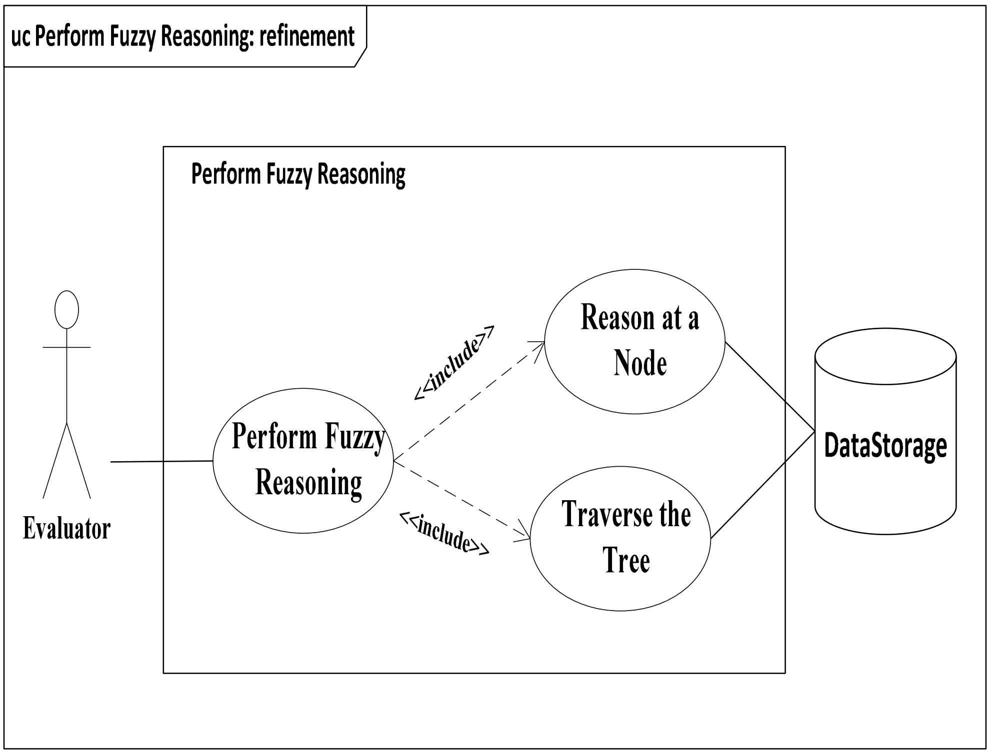


Fig. 20 Refinement of use case Perform Fuzzy Reasoning

Fig. 20 presents the decomposition of use case Perform Fuzzy Reasoning into two sub-use cases: *Travers the Tree* and *Reason at a Node*. The first one is implemented by Level Order Traversal algorithm to traverse a fuzzy AND tree. Second is implemented applying combination of Larsen implication and fuzzy aggregation techniques. We call it semantic derivation.

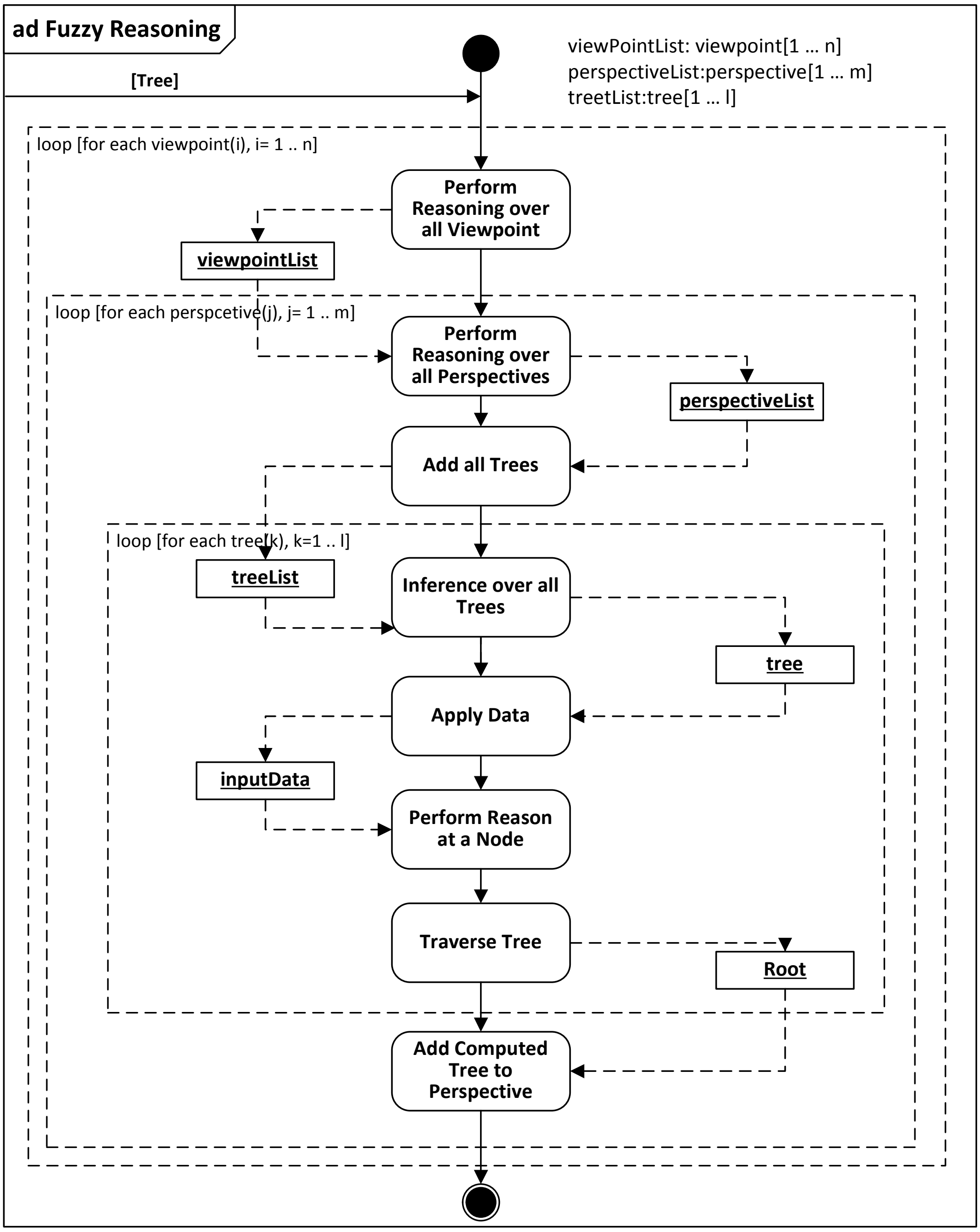


Fig. 21 Implementation of Fuzzy Reasoning engine

The implementation of the whole fuzzy reasoning engine is described by the activity diagram presented in Fig. 21**.** This diagram describes the implementation of algorithms described in Section 6.4.1. The computation process is embedded in three cycles for all viewpoints, for all perspectives in each viewpoint, and for all fuzzy AND trees in each perspective. For each tree, he value (i.e. a fuzzy number) is inferred.

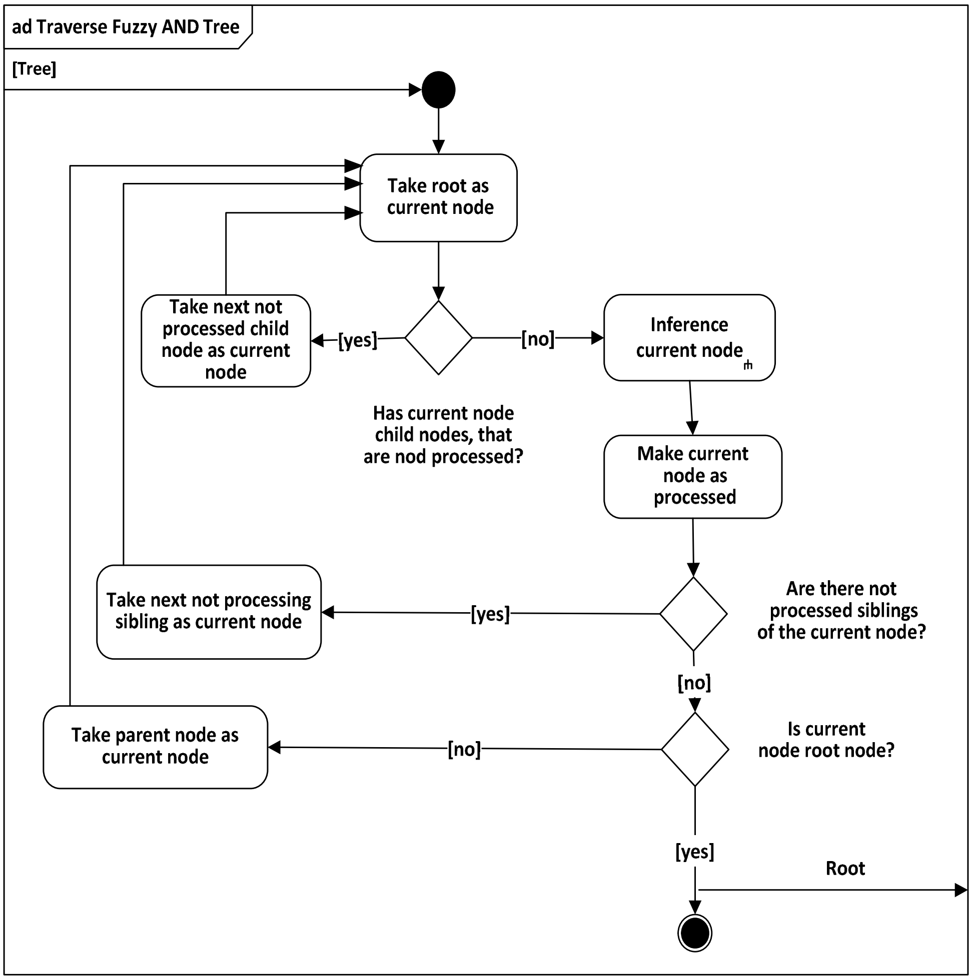


Fig. 22 Implementation of fuzzy AND tree traversal algorithm

Fig. 22 presents fuzzy AND tree traverse activity diagram. The process starts from the root of the three and proceeds down until the left leaf is reached. Then the reasoning process starts. It proceeds using the bottom-up level order traversal algorithm of its nodes' values. (i.e. from left to right, level by level from leaf to root). At each upper level node the semantic derivation of its value is performed applying Larsen implication for each child node and aggregating all obtained values. This process is described by the activity diagram presented in Fig. 23.

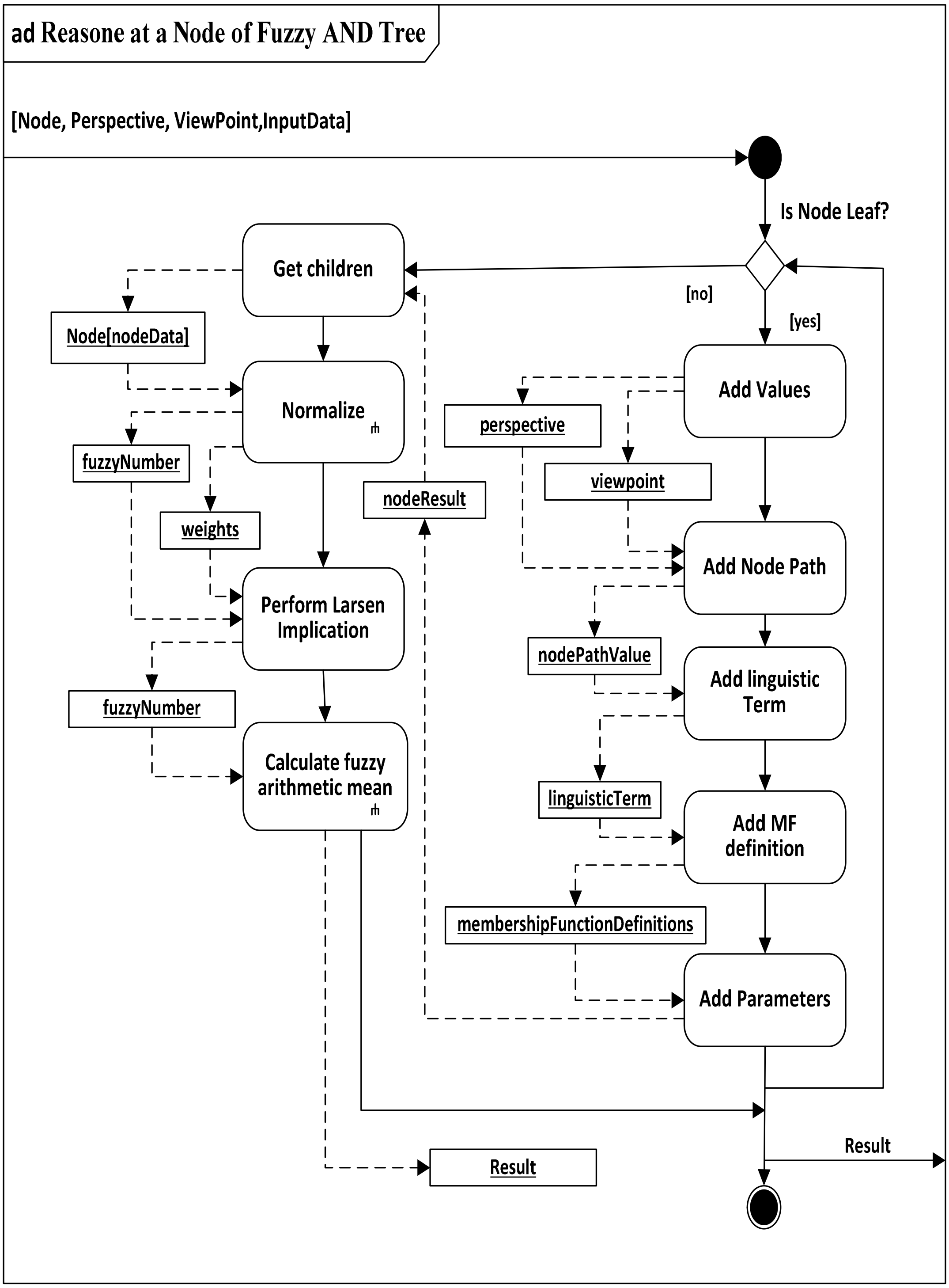


Fig. 23 Reasoning at a node of fuzzy AND tree

6.5.5.Implementation of Use Cases *Perspective Aggregation* and *Viewpoint Aggregation*

Fig. 24 presents the decomposition of use case *Aggregation* into two sub-use cases: *Aggregate Perspectives* and *Aggregate Viewpoints*.

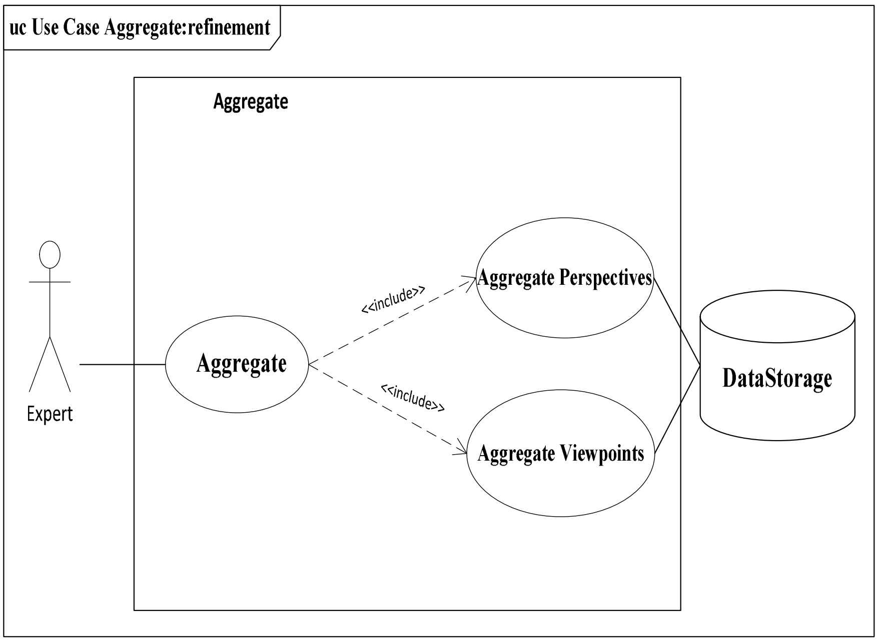


Fig. 24 View Aggregation Subsystem

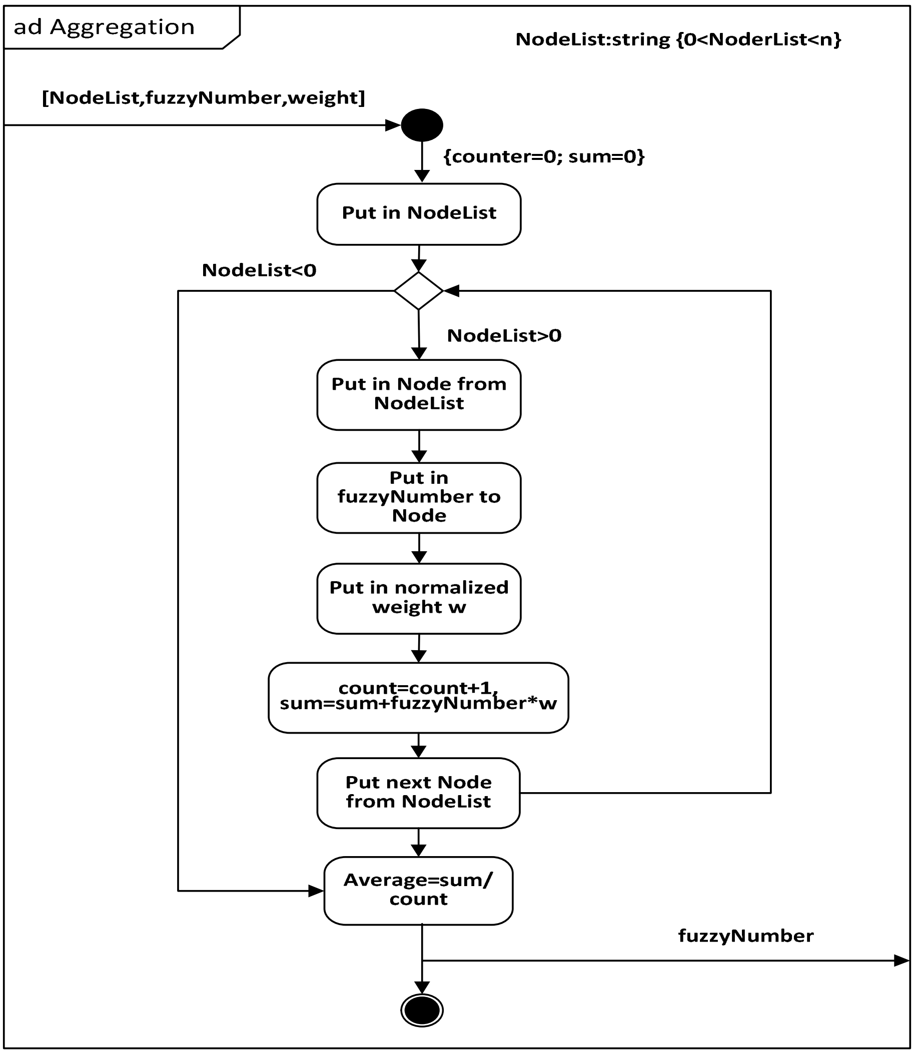


Fig. 25 Viewpoints aggregation activity diagram

Each sub-use case is implemented by its own subsystem (see Fig. 12).

The activity diagram presented in Fig. 25 describes the implementation of viewpoints aggregation algorithm. The perspectives aggregation algorithm differs in that that it has on additional (first) steps, which forms linguistic vector for each perspective from values of top-level quality attributes.

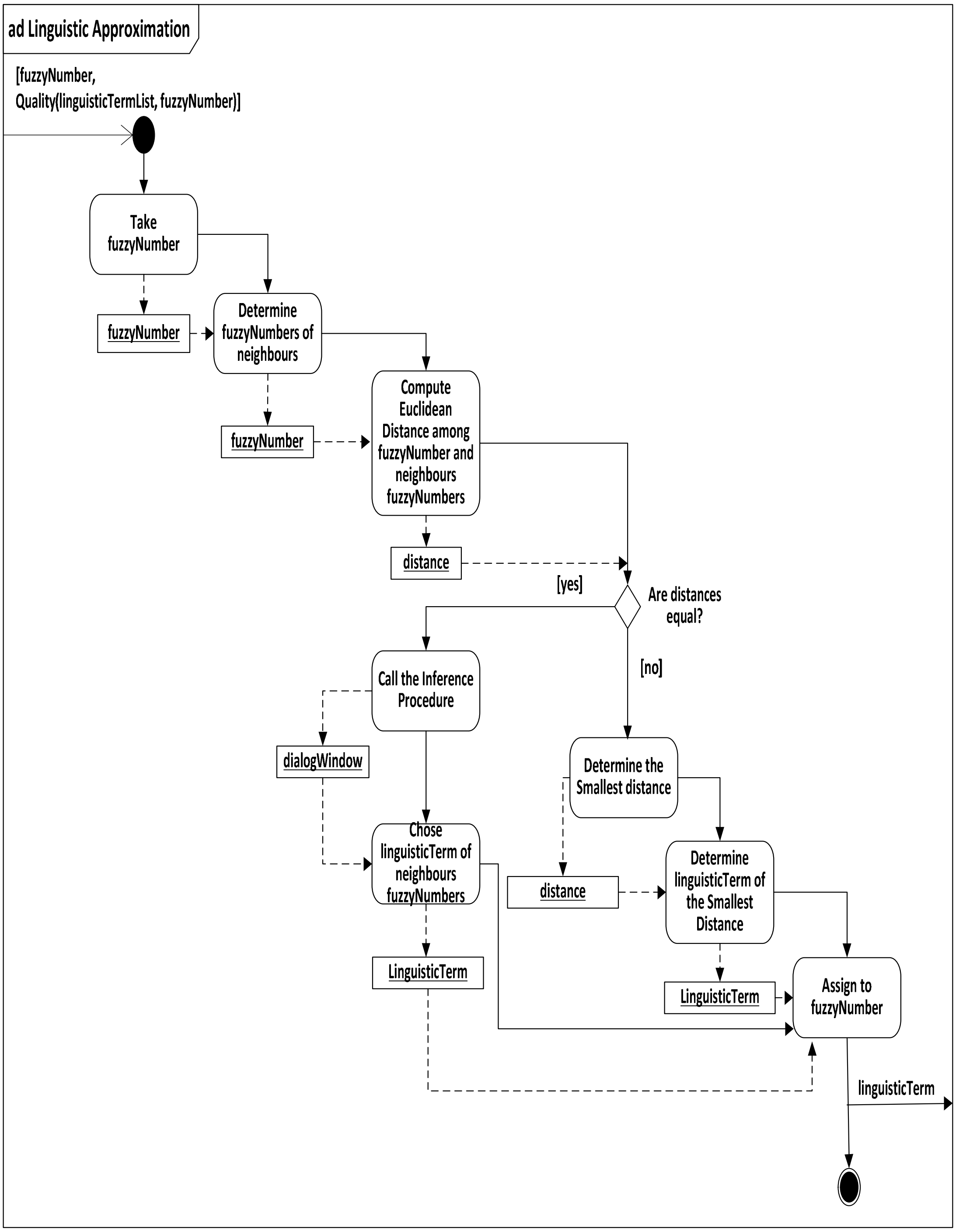


Fig. 26 Linguistic approximation activity diagram

After aggregation of all perspectives in each viewpoint and aggregation of all viewpoints, the linguistic approximation process follows (see

Fig. 26). The fuzzy numbers are the input data, and the linguistic values are the output data. Fuzzy numbers are approximated using Euclidean distance as a similarity relation. If a conflict arises (i.e. the fuzzy number can be approximated by two linguistic terms), the interactive procedure is activated and the decision should be made by an expert.

6.6. Conclusions

The main conclusions of this chapter are as follows:

1. Developed conceptual framework of concepts associated with fuzzy trees has allowed to formalize QoS_{EBS} planning model and to develop problem-oriented QoS_{EBS} quality model suited to solve QoS_{EBS} planning problem. Thus proving hypothesis H4.
2. The designed ensemble of collaborating algorithms combines problem fuzzification, interactive fuzzy balancing, fuzzy reasoning, linguistic approximation, and fuzzy aggregation algorithms. It means that the hypotheses H7 was proved.
3. The UML tool was successfully used for implementation of the Quality planning system, which involves ensemble of collaborative algorithms.

Chapter 7 Experimental Research

This chapter presents four controlled experiment, performed with the data of four concrete case studies. The aim of these case studies is to demonstrate the suitability of the proposed algorithms fuzzy reasoning, linguistic approximation, perspective aggregation, and viewpoint aggregation to solve the QoS_{EBS} planning problem. **Section 1** describes the experimental design of this research. **Section 2** provides four case studies, which also correspond to the demonstrative one. **Section 3** introduces the settings of experimental research. **Section 4** presents the observations and findings of these use cases. **Section 5** describes the threats of validity. Finally, in **Section 6** presents the conclusions.

7.1. Experimental Design

According to [245], experimental design is defined as design of experiments which "*is an efficient procedure for planning experiments so that the data obtained can be analysed to yield valid and objective conclusions*", i.e. "*the laying out of a detailed experimental plan in advance of doing the experiment*". Therefore, it refers to the plan, structure, and strategy of experiment, and guides the whole experimental process in so a way that to obtain results to yield conclusions of research questions. The experimental research focuses on the question "*Is the proposed approach acceptable as a whole?*", "*Is the ensemble of QoS_{EBS} planning methods and algorithms computational correct?*" The research results of the dissertation were evaluated, using case study controlled experiments, i.e. case study controlled methodology [51,52]. The aims of these experiments were to prove the acceptability of the proposed approach as a whole and computational correctness of the ensemble of QoS_{EBS} planning methods and algorithms.

The hypothesis H6 was justified by an evidence-based inductive reasoning procedure [8] that refers to specific observation, which moves toward a generalization. According to [246], "*Inductive reasoning begins with observations that are specific and limited in scope, and proceeds to a generalized conclusion that is likely, but not certain, in light of accumulated* *evidence*." The external validity of this experimental research was evaluated but the statistical validity was ignored.

The hypothesis H6 justified also by disproof by counterexample [7] that is it-self naturally refutation of the universal statements or by validation and verification procedures.

A hypothesis H7 was justified by a case-based controlled experiment [9] that "*is a viable substitute for the actual usage scenario*" [247] which obtained results were generalized based on logical inference.

The black box process model of the Design of Experiments (DOE) [245] was used as efficient procedure for the planning experiment. The schematic for the QoS_{EBS} planning process is shown in Fig. 27.

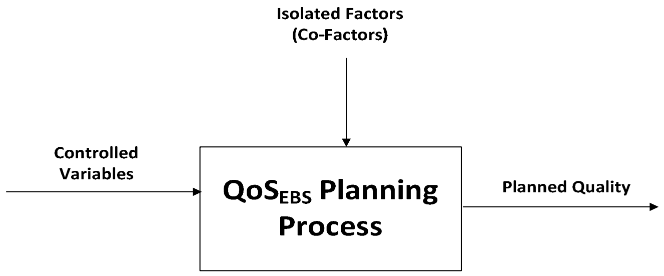


Fig. 27 Schematic for the Quality Planning Process

The QoS_{EBS} planning process includes such factors [245]:

- **Isolated factors** (co-factors). The QoS_{EBS} planning process includes the isolated factors, which are fixed, unchanging, and not manipulated during the experiment by the expert. The influence of this kind of factors on the experimental research results is indirect and inessential, i.e. don't have influence for the research results.
- **Controlled variables**. The QoS_{EBS} planning process is exposed by controlled variables (for example, linguistic terms, MFs, weights) varied at by will by the evaluator. The influence of this kind of factors on the experimental research results is direct and essential.
- **Latent variables**. The third kind of the process affected parameters is so-called latent variables (for example, time), which are not directly observed but can influence results of the measurements. The existence of the latent parameters in QoS_{EBS} planning process is not known.

Threats to validity [248] of experimental results using case-based experimental research can be performed taking into account:

- **Construct validity** is the degree to which the results of experiment match the theoretical expectations or "*the degree to which a test measures what it claims, or purports, to be measuring*". Evaluating the construct validity, the question "How well are defined the theoretical ideas, performed mapping of concepts and relations?" should be answered.
- **Internal validity** reflects the extent to which a causal conclusion based on the results of the experiment is warranted.
- **External validity** is the extent to which the results of a case study can be generalized to other QoS_{EBS} planning situations?

The objective of the experimental research was to justify the hypotheses H6 and H7, to demonstrate the correctness and acceptance of stated hypotheses by some evidence. For this reason, four case-studies were performed – fuzzy reasoning, linguistic approximation, perspectives aggregation and viewpoints aggregation. In experimental research, the settings for all case studies, in which the research is performed, are isolated by researcher (the "laboratory"). All required code for framework has been written in C# programming language. Microsoft Visual Studio 2013 integrated development environment have been used for developing and testing the framework. All measurements have been done on computer with Intel® Core™ i7-2600 @ 3.40 GHz processor, 8,00 GB of RAM, Microsoft Windows 7 Professional operating system, using built-in tools of Microsoft Visual Studio 2013 integrated development environment and C# programming language. The design results are documented using UML-like notation. The API technical documentation is generated using documentation compiler from managed class libraries Sandcastle (see in Appendix C). The threats to validity for each case study were defined and discussed. Finally, the generalization and conclusions were pointed out.

7.2. Case Studies

Four case studies were presented and conducted: case study of the fuzzy reasoning, case study of the linguistic approximation, the case study of perspectives integration, and case study of the viewpoints integration. The use cases also correspond to the demonstrative one. The purpose of these performed case studies was to demonstrate the correctness and acceptance of the proposed algorithms. The case-based controlled experiment methodology [51,52,9] was used for justification of H6 and H7 hypothesis. The obtained results of case studies are considered as fulfilled if the criterion of construct/internal/external validity is reasonable argued. If the threats to validity could not be controlled, it must be noted. One kind of controlled variable are manipulated and tested.

7.2.1. Case Study 1: Fuzzy Reasoning

Through experimentation with fuzzy reasoning algorithm, the set X of linguistic terms, which correspond to the fuzzy numbers, assigned to the attributes of QoS characteristics, for selected tree of the quality attributes, is generated. Fuzzy implication with semantic derivation techniques was combined in the fuzzy reasoning process. The treatment of controlled variables was taken: 1) all chosen linguistic terms are of the same kind of linguistic terms; 2) all chosen linguistic terms are defined in randomize way. Weights, shape of MFs are isolated factors that, during the experiment, are fixed. The set X of fuzzy numbers, assigned to the attributes of QoS, is controlled variable that is defined and for each experiment of treatment is changed.

Appendix D with the detailed results of this case study is presented in the CD that is attached to the dissertation.

7.2.2. Case Study 2: Linguistic Approximation

This case study analyses linguistic approximation issue in QoS_{EBS} planning process and describes in details its inferences in the tree structures. The objective of this case study is to show the principal feasibility of linguistic approximation. Through experimentation with linguistic approximation algorithm, the set X of fuzzy numbers, assigned to the attributes of QoS, was generated. These numbers were approximated by terms of the given linguistic variable Quality. Two treatments of controlled variable were taken: 1) approximated term can be chosen unambiguously; 2) a choice of two possible linguistic terms. Weights, shape of MFs are isolated factors that, during the experiment, are fixed. The set X of fuzzy numbers, assigned to the attributes of QoS, is controlled variable that is defined and for each experiment of treatment is changed.

Appendix E with the detailed results of this case study is presented in the CD that is attached to the dissertation.

7.2.3. Case Study 3: Perspectives Aggregation

Through experimentation with perspective aggregation algorithm, it was intended to check the meaningfulness of weighted mean formula of fuzzy numbers for the aim to aggregate the linguistic vectors of different perspectives. Weights, describing the importance of perspective in the context of the developed QoS_{EBS}, are controlled variable. The treatment of controlled variable was taken: the aggregation of perspectives within the limits of one selected viewpoint by weights changing for each QoS characteristic in each perspective. Linguistic terms, shape of MFs are isolated factors that, during the experiment, are fixed.

Appendix F with the detailed results of this case study is presented in the CD that is attached to the dissertation.

7.2.4. Case Study 4: Viewpoints Aggregation

Similarly, it was experimented with the aggregation algorithm of the viewpoints linguistic vectors. Through experimentation with viewpoint aggregation algorithm, it was intended to check the meaningfulness of weighted mean formula of fuzzy numbers for the aim to aggregate the linguistic vectors of different viewpoints. Weights, describing the importance of each of viewpoint in the context of the developed QoS_{EBS}, are controlled variable. The treatment of controlled variable was taken: change weights for each viewpoint.

Appendix G with the detailed results of this case study is presented in the CD that is attached to the dissertation.

7.3. Threats to Validity

The results of experiment were considered to meet the expectations of experimental research, if the construct validity, internal validity, and external validity of these results are showed. These threats to validity of experimental results are examined:

1. Construct validity: the obtained results are inconsistent with theoretical expectations, i.e. the algorithms do not give the results that were expected.
2. Internal validity: a) controlled variables and results are related in any other (e.g. correlation), but not a causal relationship.
3. External validity: examined causal relationship is only valid for those concrete cases to which the data has been experimented, thus, the experimental results can't be generalized (i.e. different results are obtained with another data).

7.4. Conclusions

The main conclusions of this chapter are as follows:

1. In none one of performed experiments are not found the threats of empirical validity, i.e. all results of experiments were consistent with theoretical expectations.
2. The controlled variables and results in all experiments are related with causal relationship.
3. The experiments did not reveal any specifics to used data that can damage the external validity.

4. The ensemble of collaborating algorithms suited to solve QoS_{EBS} planning problem is developed. It means that the hypotheses H7 was proved.
5. Trying fuzzy reasoning algorithm approved H6 hypothesis, that fuzzy reasoning formalism that combines semantic derivation and aggregation is acceptable for inference in tree structures, which describe the hierarchy of QoS_{EBS} properties, because it meets all functional requirements.

Conclusions

The main conclusions of the dissertation are as follows:

1. A fuzzy inference-based approach to the planning quality of enterprise business services enables us to develop a conceptual framework suitable to define QoS_{EBS}, taking into account different understandings of the concept '*quality*', to describe QoS_{EBS} planning problem in a formal way as well as to describe QoS_{EBS} quality model and planning algorithms inspired by this model.
2. In order to reason about linguistic terms in tree structures, which describe the hierarchies of QoS_{EBS} quality attributes, the most suitable is the formalism, which combines semantic fuzzy derivation and fuzzy aggregation techniques. By semantic derivation we mean Larsen implication interpretation in the semantic model[9] defined in fuzzy number domain.
3. It is impossible to fuzzify the QoS_{EBS} planning problem in this same way for any Enterprise Business Service. For this end, a methodology that guides the fuzzification process for any EBS is required. The dissertation shows the possibility to develop such methodology.
4. The designed ensemble of collaborating interactive fuzzy balancing, fuzzy reasoning, linguistic approximation, and fuzzy aggregation algorithms is sufficient to the QoS_{EBS} planning problem.
5. The object-oriented software system architecture is sufficient to implement this ensemble of collaborative algorithms.

[9] The term *model* here means model of linguistic logic.

References

1. Benbernou, S., Brandic, I., Cappiello, C., Carro, M., Comuzzi, M., Kertesz, A., Kritikos, K., Parkin, M., Pernici, B., Pleban, P.I. (2010). Modeling and Negotiating Service Quality. In : Service Research Chalenges and Solutions for the Future Internet: S-Cube - Towards Engineering, Managing and Adapting Service-Based Systems. LNCS 6500. Springer, 157-209.
2. Abramowicz, W., Hofman, R., Suryn, W., Zyskowski, D. (2008). SQuaRE based Web Services Quality Model. In Ao, S., Castillo, O., Douglas, C., Feng, D.D., Lee, A-J. (Eds.). Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2008), 19-21 March 2008, Hong Kong.LNECS 2168 I. Newswood Limited, Hong Kong, 827-835
3. Chomsky, N. (2012). The Science of Language: Interviews with James McGilvary. Cambridge University Press, Cambridge.
4. Speaks, J. (2014). Theories of Meaning. In Zalta, E.N. (Ed.). The Stanford Encyclopedia of Philosophy (Fall 2014 Edition). Stanford University.
5. Dvorak, A. (1999). On linguistic approximation in the frame of fuzzy logic deduction. Soft Computing, 3(2), 111-115.
6. Shealy, C.N. (2005). Justifying the Justification Hypothesis: Scientific-Humanism, Equilintegration (EI) Theory, and the Beliefs, Events, and Values Inventory (BEVI). Journal of Clinical Psychology, 61(1), 81–106.
7. Pahl, P.J., Damrath, R. (2012). Proofs and axioms. In : Mathematical Foundations of Computational Engineering: A Handbook, Volume II. Springer-Verlag, Berlin Heidelberg, 26-30.
8. Gray, D.E. (2013). Doing Research in the Real World, 3rd edition. Sage Publications.
9. Ko, A.J., LaToza, T.D., Burnett, M.M. (2015). A practical guide to controlled experiments of software engineering tools with human participants. Empirical Software Engineering, 20(1), 110-141.
10. Snijders, T.A.B. (2001). Hypothesis Testing: Methodology and Limitations. In Smelser, L.J., Baltes, P.B. (Eds.). International Encyclopedia of the Social & Behavioral Sciences, vol.10. Elsevie-Pergamon, 7121-7127.
11. Bickman, L., Rog, D.J. (2008). Applied Research Design. A Practical Approach. In Bickman, L., Rog, D.J. (Eds.). The Sage Handbook of Applied Social Research Methods.Second edition. Sage publications, 3-43.
12. Dietterich, T.G. (1990). Editorial: Exploratory Research in Machine Learning. Machine Learning, 5(1), 5-9.
13. Besson, J., Čaplinskas, A. (2009). Conceptual Analysis of the Notion of Quality of Services. Lietuvos matematikų draugijos darbai, 50, 357-361.
14. Besson, J., Čaplinskas, A. (2009). Quality of services in the context of internet of services. Informacijos mokslai, (50), 248-256.
15. Besson, J., Čaplinskas, A. (2010). QoS-aware composition of enterprise system‘s

components: constraint logic programming approach. Informatica, 21(4), 487-504.

16. Leite, J.C.S.P. (1989). Viewpoint Analysis: A Case Study. In : IWSSD '89 Proceedings of the 5th international workshop on Software specification and design, 111-119.
17. Sommerville, I., Sawyer, P. (1997). Viewpoints: principles, problems and a practical approach to requirements engineering. Annals of Software Engineering, (3), 101–130.
18. Leite, J. (1989). Viewpoint Analysis: A Case Study. In: Proceedings of the 5th international workshop on Software specification and design IWSSD '89. ACM, New York, 111-119.
19. Leite, J.C.S.P. (1988). Viewpoint resolution in requirements elicitation. Ph.D. thesis, Dept. Computer Sci., Univ. California, Irvine.
20. Letsholo, K., Chioasca, E.-V., Zhao, L. (2012). An Integration Framework for Multi-Perspective Business Process Modeling. In: Proceedings of the 2012 IEEE Ninth International Conference on Services Computing. IEEE Computer Society, Washington, DC, USA, 33-40.
21. Frank, U. (2002). Multi-perspective Enterprise Modeling (MEMO) Conceptual Framework and Modeling Languages. In Sprague Jr., R.H. (Ed.). Proceedings of the Hawaii International Conference on System Sciences (HICSS-2002). Los Alamitos, CA. IEEE Computer Society, Washington, DC, USA, 1258–1267.
22. Verjus, H., Cıımpan, S.:Razavizadeh, D. (2010). BeeEye: A Framework for Constructing Architectural Views. In Babar, M.A., Gorton, I. (Eds.). Software Architecture. 4th European Conference, ECSA 2010, Copenhagen, Denmark, August 23-26, 2010 LNCS 6285. Springer-Verlag, Berlin, Heidelberg, 376-383.
23. Thanitsukkarn, T. (1999). Multiperspective Development Environment for Configurable Distributed Applications. PhD Thesis, Imperial College of Science, Technology and Medicine, University of London, Department of Computing, London.
24. Yu, E. (1995). Modelling Strategic Relationships for Process Reengineering. PhD thesis, University of Toronto, Toronto.
25. Yu, E.S.K. (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97) Jan. 6-8, 1997, Washington D.C., USA. IEEE, 226-235.
26. Horkoff, J., Yu, E. (2010). A framework for iterative, interactive analysis of agent-goal models in early requirements engineering. In: Castro, J., Franch, X., Mylopoulos, J., Yu, E. (Eds.). Proceedings of the 4th International i* Workshop (iStar-2010), Hammamet, Tunisia, 7–8 June 2010, 52-56.
27. Reed, J.G., Baxter, P.M. (2006). Biblographic research. In: Leong, F.T.L., Austin, J.T. (Eds.). The Psychology Research Handbook: A Guide for Graduate Students and Research Assistants 2nd edn. Sage Publications, 41-58.
28. Creswell, J.W. (2003). Research Design:Qualitative, Quantitative, and Mixed Methods Approaches. 2nd edn. Sage publications.
29. Kitchenham, B. (2004). Procedures for Performing Systematic Reviews. Joint Technical Report TR/SE-0401.

30. Kitchenham, B. et al. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report, Ver. 2.3 EBSE Technical Report. EBSE.
31. Kitchenham, B. (2009). Systematic Literature Reviews in Software Engineering - a Systematic Literature Review. Information and Software Technology, 51(1), 7-15.
32. Kitchenham, B. (2004). Evidence-Based Software Engineering. In : Proceedings of the 26th International Conference on Software Engineering, (ICSE'04), Edinburgh, UK, May 23-28, 2004. IEEE Computer Society, Washington, DC, USA, 273-281.
33. Biolchini, J., Mian, P.G., Natali, A.C.C., Travassos, G.H. (2005). Systematic Reviews in Software Engineering, Technical Report TR-ES 679/05.
34. Barnett-Page, E., Thomas, J. (2009). Methods for the synthesis of qualitative research: a critical review. BMC Medical Research Methodology, 9-59.
35. Evans, D. (2002-2003). Systematic reviews of interpretive research: interpretive data synthesis of processed data. Australian Journal of Advanced Nursing, Vol. 20, 22-26.
36. Woolf, S. (1992). Practice guidelines, a new reality in medicine. II: Methods of developing guidelines. Archives of Internal Medicine, 5(152), 946-952.
37. Noblit, G.W., Hare, R.D. (1998). Meta-ethnography: synthesising qualitative studies. Sage Publications, Newbury Park, California.
38. Dixon-Woods, M., Agarwal, Sh., Jones, D., Young, B., Sutton, A. (2005). Synthesising qualitative and quantitative evidence: a review of possible methods. Journal of Health Services Research & Policy, 10(1), 45-53.
39. Denyer, D., Tranfield, D. (2006). Using qualitative research synthesis to build an actionable knowledge base. Management Decision, 44(2), 213-227.
40. Svejvig, P. (2009). Literature Review of Enterprise Systems Research Using Institutional Theory: Towards a Conceptual Model, WORKING PAPER 2009-01. University of Aarhus, Aarhus School of Business, Department of Management.
41. Hagen-Zanker, J., Mallett, R. (2013). How to do a rigorous, evidence-focused literature review in international development. A Guidance Note. ODI Working Paper. Overseas Development Institute, London.
42. Crnkovic, G.D. (2010). Constructive Research and Info-Computational Knowledge Generation. Model-based reasoning in science and technology, Studies in Computational Intelligence 314/2010, 359-380.
43. Lukka, K. (2003). The constructive research approach. In: Ojala, L., Hilmola, O-P. (Eds.). Case study research in logistics, Series B 1: 2003. Publications of the Turku School of Economics and Business Administration, 83-101.
44. Sipser, M. (2012). Types of Proof. In: Introduction to the Theory of Computation 3rd edn. Cengage Learning, Boston, USA, 21-28.
45. Steiner, E. (1988). Methodology of Theory Building. Educology Research Associates, Sydney.
46. Colton, S. (2000). Assessing Exploratory Theory Formation Programs. In: Proceedings of the AAAI-2000 Workshop on new research directions in machine learning. URL: http://ccg.doc.gold.ac.uk/papers/colton_aaai00w.pdf

47. Wacker, J.G. (1998). A definition of theory: research guidelines for different theory-building research methods in operations management. Journal of Operations Management, 16(4), 361–385.
48. Jabareen, Y. (2009). Building a Conceptual Framework: Philosophy, Definitions, and Procedure. International Journal of Qualitative Methods, 8(4), 49-62.
49. Meredith, J. (1993). Theory Building through Conceptual Methods. International Journal of Operations & Production Management, 13(5), 3-11.
50. Machado, A., Silva, F.J. (2007). Toward a Richer View of the Scientific Method: The Role of Conceptual Analysis. American Psychologist, 62(7), 671–681.
51. Pfleeger, S.L. (1994). Design and Analysis in Software Engineering: Part 1: The Language of Case Studies and Formal Experiments. Software Engineering Notes, 19(4), 16-20.
52. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D. (). Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J., Sjøberg, D.I.K. (Eds.). Guide to Advanced Empirical Software Engineering. Springer London, London, 285-311.
53. Salkind, N.J. (). Encyclopedia of Research Design. SAGE Publications, Thousand Oaks, California.
54. Garvin, D.A. (1988). Managing Quality: The Strategic and Competitive Edge. Simon&Schuster.
55. Garvin, D.A. (1984). What Does 'Product Quality' Really Mean? Sloan Management Review, 26(1), 25-43.
56. Garvin, D.A. (1987). Competing on the eight dimensions of quality. Harvard Business Review, 65(6), 101–109.
57. Edvardsson, B. (1998). Research and concepts: service quality improvement, Managing Service Quality, 142-149.
58. Edvardsson, B. (2005). Service quality: beyond cognitive assessment, Managing Service Quality, 127-31.
59. Evans, J., Lindsay, W. (2008). Managing for Quality and Performance Excellence,7th ed., Southwestern, Thomson.
60. Nankivel, J. (2011). Quality Perspectives in Project Management. pmStudent, (2011 [cited 2012 February 14]).
61. Ross, J.E. (1994). Total Quality Management: Text, Cases, and Readings. Kogan Page.
62. Kitchenham, S., Pfleeger, S.L. (1996). Software quality: the elisive target. IEEE Software, 13(1), 12-21.
63. Pirsig, R.M. (2006). Zen and the Art of Motorcvcle Maintenance Reprint edition. HarperTorch, New York.
64. Lacewing, M. (2014). Plato’s theory of Forms. In: Routledge Taylor & Francis Group ALP. Available at: http://cw.routledge.com/textbooks/philosophy/downloads
65. Lohrman, M.J. (2015). Business Process Quality Management, Dissertation zur Erlangung des Doktorgrades, Dr. rer. nat., Universität Ulm, Ulm, Germany.
66. Broh, R.A. (1982). Managing Quality for Higher Profits: A Guide for Business Executives and Quality Managers. McGraw Hill Higher Education , Columbus, USA.

67. Boehm, B. (2006). Value-Based Software Engineering: Overview and Agenda. In: Biffl, S., Aurum, A., Boehm, B., Erdogmus, Grünbacher, P. (Eds.). Value-Based Software Engineering, Part I., Springer, Berlin Heidelberg, 3-14.
68. Juran, J.M., Godfrey, A.B. (1998). Juran's Quality Handbook, 5th edn. McGraw-Hill.
69. ISO/IEC (2001). Software engineering — Product quality. Part 1:Quality model. International Standard 9126-1, 1st edn., ISO/IESC, Geneva.
70. Balin, S, Giard, V. (2006). A process oriented approach to the service concepts. In: Proceedingsof the 2006 International Conference on Service Systems and Service Management (ICSSSM'06), 25-27 October 2006, Troyes, France I. IEEE, 785 - 790.
71. Kotler, Ph., Dubois, B. (2003). Marketing Managemen, 11th edn., Pearson Education, Paris.
72. Gronröos, Ch. (1988). Marketing Services: The Case of the Missing Product. Journal of Business and Industrial Marketing, 13(4), 322-338.
73. Sassen, A.M., Macmillan, Ch. (2005). The service engineering area: An overview of its current state and a vision of its future, versio 1.0. European Commission, Information Society and Media Directorate-General.
74. Gadrey, J. (2000). The Characterization of Goods and Services: An Alternative Approach. Review of Income and Wealth, 46(3).
75. Laroche, M., Bergeron, J., Goutaland, Ch.A. (2001). Three-Dimensional Scale of Intangibility. Journal of Service Research, 4(1), 26-38.
76. Moeller, S. (2010). Characteristics of services – a new approach uncovers their value. Journal of Services Marketing, 24(5), 359-368.
77. Edvardsson, B. (2005). Service portraits in service research: a critical review. International Journal of Service Industry Managemen, 16(1), 107-121.
78. Hofacker, Ch.F. (2007). E-Services: A Synthesis and Research Agenda. Journal of Value Chain Management, 1(1/2), 13-44.
79. Kang, G.-D., James, J. (2004). Service quality dimensions: an examination of Grönroos's service quality model. Managing Service Quality, 14(4), 266–277.
80. Grönroos, Ch. (1984). A Service Quality Model and its Marketing Implications. European Journal of Marketing, 18(4), 36-44.
81. Parasuraman, A., Zeithaml, V.A., Berry, L.L. (1985). A Conceptual Model of Service Quality and Its Implications for Future Research. Journal of Marketing, 49(4), 41-50.
82. Haywood-Farmer, J., Downing, M. (1986). Controlling Service Quality. Business Quarterly, 50(4), 62-67.
83. Brogowicz, A.A., Delene, L.M., Lyth, D.M. (1990). A synthesised service quality model with managerial implications. International Journal of Service Industry Management, 1(1), 27-44.
84. Cronin, J.J.Jr., Taylor, S.A. (1992). Measuring Service Quality: A Reexamination and Extension. Journal of Marketing, 56, 55-68.
85. Mattsson, J. (1992). A service quality model based on ideal value standard. International Journal of Service Industry Management, 3(3), 18-33.
86. Teas, R.K. (1994). Expectations: a comparison standard in measuring service

quality: an assessment of a reassessment. Journal of Marketing, 58(1), 132-139.

87. Berkley, B.J., Gupta, A. (1994). Improving service quality with information technology. International Journal of Information Management, 14(2), 109-121.
88. Dabholkar, P.A., Bagozzi, R.P. (2002). An Attitudinal Model of Technology - Based Self - Service: Moderating Effects of Consumer Traits and Situational Factors. Journal of the Academy of Marketing Science, 30(3), 184-201.
89. Spreng, R.A., Mackoy, R.D. (1996). An empirical examination of a model of perceived service quality and satisfaction. Journal of retailing, 72(2), 201-214.
90. Philip, G., Hazlett, S.A. (1997). The measurement of service quality: a new P-C-P attributes model. International Journal of Quality & Reliability Management, 14(3), 260-86.
91. Sweeney, J.C., Soutar, G.N., Johnson, L.W. (1997). Retail service quality and perceived value. Journal of Consumer Services, 4(1), 39-48.
92. Oh, H. (1999). Service quality, customer satisfaction and customer value: a holistic perspective. International Journal of Hospitality Management, 18, 67-82.
93. Dabholkar, P.A., Shepherd, C.D., Thorpe, D.I. (2000). A comprehensive framework for service quality: an investigation of critical conceptual and measurement issues through a longitudinal study. Journal of Retailing, 76(2), 131-139.
94. Frost, F.A., Kumar, M. (2000). INTSERVQUAL: an internal adaptation of the GAP model in a large service organization. Journal of Services Marketing, 14(5), 358-377.
95. Soteriou, A.C., Stavrinides, Y. (2000). An internal customer service quality data envelope analysis model for bank branches. International Journal of Bank Marketing, 18(5), 246-252.
96. Broderick, A.J., Vachirapornpuk, S. (2002). Service quality in internet banking: the importance of customer role.. Marketing Intelligence & Planning, 20(6), 327-335.
97. Zhu, F.X., Wymer, W.J., Chen, I. (2002). IT-based services and service quality in consumer bankingInternational. Journal of Service Industry Management, 13(1), 69-90.
98. Santos, J. (2003). E-service quality: a model of virtual service quality dimensions. Managing Service Quality, 13(3), 233-246.
99. Seth, N., Deshmukh, S.G., Vrat, P. (2005). Service quality models: a review. International Journal of Quality & Reliability Management, 22(9), 913-949.
100. Hajek, P. (2010). Fuzzy Logic. In Zalta, E.N. (Ed.). The Stanford Encyclopedia of Philosophy, Fall 2010 Edition. Center for the Study of Language and Information, Stanford UniversityAccsessible: http://plato.stanford.edu/archives/fall2010/entries/logic-fuzzy/
101. Kowalczyk, R. (1999). On numerical and linguistic quantification in linguistic approximation. In: Systems, Man, and Cybernetics. IEEE SMC'99 Conference Proceedings, Vol. 5, 326-331.
102. Papazoglou, M., Georgakopoulos, D. (2003). Service-Oriented Computing. Communications of the ACM, 46(10), 25-28.
103. Szyperski, C. (2002). Component Software: Beyond Object-Oriented Programming, 2nd edn. ACM Press and Addison-Wesley, New York, NY.
104. Raje, R.R. (2001). UMM: Unified Meta-object Model for Open Distributed

Systems. In the. In : Proceedings of 4th IEEE International Conference on Algorithms and Architecture for Parallel Processing, (ICA3PP'2000), Hongkong, 2000, 454-465.

105. Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology and Design. Prentice Hall.
106. Erl, T. (2008). SOA Principles of Service Design. Prentice Hall.
107. Bianco, P., Kotermanski, R., Merson, P. (2007). Evaluating a Service-Oriented Architecture. Technical Report CMU/SEI-2007-TR-015, ESC-TR-2007-015, Software Engineering Institute, Carnegy Mellon.
108. Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R. (2005). Service-Oriented Architecture (SOA) Compass - BusinessValue, Planning, and Enterprise Roadmap. IBM Press.
109. Goikoetxea, A. (2007). Enterprise Architectures and Digital Administration: Planning, Design and Assessment. World Scientific.
110. Mueller, T, Schuldt, D, Sewald, B., Morisse, M., Petrikina, J. (2013). Towards inter-organizational Enterprise Architecture Management - Applicability of TOGAF 9.1 for Network Organizations. In: Proceedings of the Nineteenth Americas Conference on Information Systems (AMCIS 2013): HYPERCONNECTED WORLD: Anything, Anywhere, Anytime. Chicago, Illinois, August 15-17, 2013 5. Association for Information Systems (AIS), Atlanta, Georgia, USA (2013), 3648-3660.
111. Bieberstein, N., Bose, S., Walker, L., Lynch, A. (2005). Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Systems Journal, 44(4), 691-708.
112. SAP (2008). Enterprise SOA Development Handbook 1.1., SAP AG.
113. Hack, S., Lindemann, M. (2008). Enterprise SOA Roadmap. Galileo Press.
114. Tang, L., Dong, J., Peng, T., Tsai, W.-T. (2010). Modeling enterprise service-oriented architectural stylesi. Service Oriented Computing and Applications, 4(2), 81–107.
115. Allega, P.J. (2005). Architecture Framework Debates Are Irrelevant. Research G00127331. Gartner.
116. SOA-RAF (2012). Reference Architecture Fundation for Service Oriented Architecture Version 01. OASIS Committee Specification 01 (04 December 2012).
117. Hammer, M. (2003). The Agenda: What Every Business Must Do to Dominate the Decade. Crown Business , New York.
118. Papazoglou, M. (2008). Web Services: Principles and Technology. Prentice Hall.
119. Hill, J.B., Sinur, J., Flint, D., Melenovsky, M.J. (2006). Gartner’s position on business process management, 2006. In: Business Issues. Gartner, Stamford, CT.
120. Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, Ch., Bloch, B., Curbera, F., Ford, M, Goland, Y., Guízar, A., Kartha, N, Liu, C.K., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P.(2007). Web Services Business Process Execution Language, Version 2.0, OASIS Standard. OASIS.
121. Ko, R.K.L., Lee, S.S.G., Lee, E.W. (2009). Business process management (BPM) standards: a survey. Business Process Management Journal, 15(5), 744-791.

122. Rosen, M., Liublinsky, B., Smith, K.T., Balcer, M.J. (2008). Applied SOA: Service-Oriented Architecture and Design Strategies. Willey, Indianapolis, IN, USA.
123. Albreshne, A., Fuehrer, P., Pasquer, J. (2009). Web Services Orchestration and Composition: Case Study of Web services Composition. Internal Working Paper no 09-03, University of Fribourg, Switzerland. Available at: https://diuf.unifr.ch/drupal/softeng/sites/diuf.unifr.ch.drupal.softeng/files/file/publications/internal/WP09-03.pdf
124. Matjaz, B.J. (2014). A Hands-on Introduction to BPEL. In: Oracle Technology Network Articles, 2009. (Accessed 2014) Available at: http://www.oracle.com/technetwork/articles/matjaz-bpel1-090575.html
125. Pourmirza, S., Dijkman, R.M. (2012). A Survey of Orchestration Engines. ICT--2011.8 Get Service Project 2012-318275 Deliverable D7.1 Available: http://getservice-project.eu/Documents2/GET%20Service%20D7_1%20Survey%20of%20Orchestration%20Engines.pdf
126. Menge, F. (2007). Enterprise Service Bus. In: Proceedings of the Free and Open Source Software Conference 2007, 1-6.
127. Chinnici, R., Moreau, J., Ryman, A., Weerawarana, S. (2007). Web Services Description Language (WSDL), Version 2.0, Part 1: Core Language. W3C Recommendation.
128. Karande, A, Karande, M., Meshram, B.B. (2011). Choreography and Orchestration using Business Process Execution Language for SOA with Web Services. IJCSI International Journal of Computer Science Issues, 8(2), 224-232.
129. Newcomer, E., Lomow, G. (2005). Understanding SOA with Web services. Addison-Wesley.
130. Chen, S. The Concept of Representational State Transfer (REST). Available at: http://ceit.uq.edu.au/system/files/fileupload/the_concept_of_representational_state_transfer.pdf
131. Fielding, R.T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Ph.D. thesis. University of California, Irvine, California, USA.
132. Sankar, K., Bouchard, S.A. (2009). Enterprise Web 2.0 Fundamentals. Cisco Press, Indeanapolis, IN, USA.
133. Chapell, D. (2007). Introducing Windows Communication Foundation. Microsoft Corporation.
134. Sadtler, C., Cuomo, G., Ganci, J., Haberkorn, M., Jones, C., Kovari, P., Griffith, K., Marhas, D., Will, R. (2005). WebSphere Product Family Overview and Architecture. Redbooks, IBM.
135. Sambeth, M. (2006). Enterprise SOA. Mastering Future Business. Presentation slides, SAP AG. Available at: http://www.aksapos.de/fileadmin/Archiv/SAP-SOA.pdf
136. Fischer, C. (2007). Enterprise SOA: General Principles, ES Bundles, ATP Check. Presentation slides, SAP AG. Available at: http://epic.hpi.uni-potsdam.de/pub/Home/TrendsAndConceptsExerciseSS2007/01_1_TrendsAndConcepts_I_Exercise.pdf
137. Kritikos, K., Plexousakis, D. (2008). Requirements for QoS-based Web Service Description and Discovery. International Journal of Web Services Practices, 3(2),

72-82.

138. Grönroos, Ch. (1982). Strategic Management and Marketing in the Service Sector. Research report 8, Swedish School of Economics and Business Administration, Helsingfors.
139. Parry, G., Newnes, L., Huang, X. (2011). Goods, Products and Services. In Macintyre, M., Parry, G., Angelis, J. (Eds.). Service Design and Delivery, Service Science: Research and Innovations in the Service Economy XXII. Springer, New York, Dordrecht, Heidelberg, London, 19-29.
140. Wolak, R., Kalafatis, S., Harris, P. (1998). An Investigation Into Four Characteristics of Services. Journal of Empirical Generalisations in Marketing Science, 3(2), 22-43.
141. Oliva, R., Kallenberg, R. (2003). Managing the transition from products to services. International Journal of Service Industry Management, 14(2), 160-172.
142. Shostack, G.L. (1984). Designing services that delive. Harvard Business Review (62).
143. Grönroos, Ch. (2000).Service Management and Marketing: A Customer Relationship Management Approach 2nd edn. John Wiley & Sons, Chichester.
144. Franch, X. (2006). On the Quantitative Analysis of Agent-Oriented Models. In Dubois, E., Pohl, K. (Eds.). CAiSE'06. Springer, Heidelberg, 495-509.
145. Kuyoro, S.O., Awodele, O., Akind, R.O., Okolie, S.O. (2012). Quality of service (Qos) issues in web services. International Journal of Computer Science and Network Security, 12(1), 94-97.
146. Menasce, D. (2002). QoS issues in Web services. IEEE Internet Computing Journal, (6), 72-75.
147. Zapater, M., Bressan, G. (2007). A Proposed Approach for Quality of Experience Assurance for IPTV. In: Proceedings of the First International Conference on the Digital Society (ICDS '07), 2-6 January 2007, Gosier, Guadeloupe. IEEE Computer Society, Washington, DC, USA, 25.
148. Wang, G., Chen, A., Wang, Ch., Fung, C., Uczekaj, S. (2004). Integrated Quality of Service (QoS) Management in Service-Oriented Enterprise Architectures. In: Proceedings of 8th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004), 20-24 September 2004, Monterey, California, USA, 21-32.
149. Farroha, B., Farroha, D.L. (2007). Policy-based QoS Requirements in a SOA Enterprise Framework-an Investigative Analysis. In: Proceedings of IEEE Military Communications Conference (MILCOM 2007), 1-7.
150. Wu, Q., Li, P. (2010). Study and Implement of Dynamic Routing Based on QoS in Enterprise Service Bus. Journal of Computational Information Systems, 6(7), 2093-2098.
151. Usmani, Sh., Azeem, N., Samreen, A. (2011). Dynamic Service Composition in SOA and QoS Related Issues. International Journal of Computer Technology and Applications, 2(5), 1315-1321.
152. Assmann, M. (2009). Model-Based Evaluation of Service-Oriented Enterprise Architectures. PhD thesis. University of Paderborn, Paderborn
153. Amini, S., Jochem, R. (2011). A Conceptual Model Based on the Fuzzy Set Theory to Measure and Evaluate the Performance of Service Processes. In :

Proceedings of the 15th IEEE International Conference on Enterprise Distributed Object Computing Workshops (EDOCW), Aug. 29-Sept. 2 2011, Helsinki. IEEE, 122-131.

154. Abramowicz, W., Hofman, R., Suryn, W., Zyskowski, D. (2008). SQuaRE based web services quality model. In: Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2008), Hong Kong, Vol. I, 827-835.
155. Abramowicz, W., Hanievicz, K., Hofman, R., Kaczmarek, M., Zykowski, D. (2009). Decomposition of SQuaRE based requirements for the needs of SOA applications. In Ao, S.-I., Huang, X., Wai, P.-K., eds. : Trends in Communication Technologies and Engineering, Lecture Notes in Electrical Engineering 33. Springe rNetherlands, 81-94.
156. ISO/IEC (2011). ISO/IEC 25010:2011 Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) -System and software quality models. International Standardisation Organisation.
157. Döweling, S., Schmidt, B., Göb, A. (2012). A Model for the Design of Interactive Systems based on Activity Theory. In: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW'12), February 11–15, 2012, Seattle, Washington, USA. ACM, New York, NY, USA, 530-548.
158. OASIS (2005). Quality Model for Web Services (WSQM -2.0). A working draft. OASIS. Available at: http://www.clip.dia.fi.upm.es/Projects/S-CUBE/papers/oasis05:WSQM-2.0.pdf
159. OASIS (2012). Web Services Quality Factors. Version 1.0.Candidate to OASIS Standard, 31 October 2012. Available at: http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cos01/WS-Quality-Factors-v1.0-cos01.pdf
160. Delor, E., Darimont, R., Rifaut, A. (2003). Software Quality Starts with the Modelling of Goal-Oriented Requirements. In: Proceedings of 16th International on Conference Software & Systems Engineering and their Applications (ICSSEA 2003), Paris, France, December 2-4,2003, 2/5-6/6.
161. Rolland, C., Prakash, N. (2000). Bridging the gap between organisational needs and ERP functionality. Requirements Engineering, 5(3), 180–193.
162. Stirna, J., Persson, A. (2007). Ten years plus with EKD: reflections from using an enterprise modeling method in practice. In: Pernici, B., Gulla, J. (Eds.). Proceedings of the 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2007), Vol. 365, 99-108.
163. Kavakli, E., Loucopoulos, P. (2004). Goal modelling in requirements engineering: analysis and critique of current methods. Information Modeling Methods and Methodologies, 102–124.
164. van Lamsweerde, A. (2001). Goal-oriented requirements engineering: a guided tour (invited paper). In: Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE 2001), Washington, 249–263.
165. Giorgini, P., Mylopoulos, J., Sebastiani, R. (2004). Simple and Minimum-Cost Satisfiability for Goal Models. In Persson, A., Stirna, J. (Eds.). CAiSE 2004 LNCS 3084. Springer, Heidelberg, 20-35.
166. Sebastiani, R., Giorgini1, P., Mylopoulos, J. (2004). Simple and Minimum-Cost Satisfiability for Goal Models. In Persson, A., Stirna, J.(Eds.). Advanced Information Systems Engineering LNCS 3084. Springer, Heidelberg, 20-35.
167. Letier, E., van Lamsweerde, A. (2004). Reasoning about Partial Goal Satisfaction

for Requirements and Design Engineering. In: Proceedings of the 12th ACM International Symposium on the Foundations of Software Engineering (FSE 2004). ACM, New York, USA, 53–62.

168. Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., Yu, E. (2010). Evaluating Goal Models within the Goal-oriented Requirement Language. International Journal of Intelligent Systems, 25(8), 841-877.

169. Giorgini, P., Nicchiarelli, E., Mylopoulos, J., Sebastiani, R. (2002). Reasoning with Goal Models. In Paccapietra, S., March, S.T., Kambashi, Y. (Eds.). Conceptual Modeling – ER'02. 21st International Conference on Conceptual Modeling Tampere, Finland, October 7–11, 2002 Proceedings LNCS 2503. Springer, Berlin. Heidelberg, 167-181.

170. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R. (2003). Formal reasoning techniques for goal models. Journal on Data Semantics(I), 1-20.

171. Horkoff, J., Yu, E. (2009). A qualitative, interactive evaluation procedure for goal- and agent-oriented models. In Yu, E., Eder, J., Rolland, C. (Eds.). Proceedings of the CAiSE'09 Forum at the 21th International Conference on Advanced Information Systems Engineering, Amsterdam, The Netherlands, 8–12 June 2009, 19-24.

172. Chung, L., Nixon, B., Yu, E. (1995). An approach to building quality into software architecture. In: Proceedings of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research (CD-ROM). IBM Press, Toronto, Ontario, Canada, 13.

173. Liu, L., Yu, E. (2004). Designing information systems in social context: a goal and scenario modelling approach. Information Systems, 29(2), 187-203.

174. Chameau, J., Santamarina, J. (1987). Membership functions I: Comparing methods of measurement. International Journal of Approximate Reasoning, 1(3), 287-301.

175. Bilgiç, T., Turksen, I.B. (1997). Elicitation of Membership Functions: How far can theory take us? In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Vol. 3, 1321-1325.

176. Lee, D. (2006). A generalized approach for analysing transportation user perception using fuzzy sets. ProQuest LLC.

177. Bilgiç, T., Türkşen, I. (2000). Measurement of Membership Functions: Theoretical and Empirical Work. In Dubois, D., Prade, H. (Eds.). Fundamentals of Fuzzy Sets 7. Springer USA,195-227.

178. Dubois, D., Ostasiewicz, W., Prade, H. (2000). Fuzzy sets: history and basic notions. In Dubois, D., Prade, H. (Eds.). Fundamentals of Fuzzy Sets. Springer Science+Business Media, LLC, New York, 21-124.

179. Aamodt, A. (1993). A case-based answer to some problems of knowledge-based systems. In Sandewall, E., Jansson, C.G. (Eds.). Proceedings of the Fourth Scandinavian Conference on Artificial Intelligence -'93, Stockholm, Sweden, 168-182.

180. Fisher, R. (1955). Statistical Methods and Scientific Induction. Journal of the Royal Statistical Society. Serie B (Methodological), 17(1), 69-78.

181. Medasani, S., Kim, J., Krishnapuram, R. (1998). An overview of membership function generation techniques for pattern recognition. International Journal of approximate reasoning, 19(3-4), 391-417.

182. Pedrycz, W., Vukovich, G. (November 2002). On Elicitation of Membership Functions. IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 32(6), 761-767.
183. Wijayasekara, D., Manic, M. (2014). Data Driven Fuzzy Membership Function Generation for Increased Understandability. In: Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2014), Beijing, China, July 6-11, IEEE, 2014 (in press).
184. Babuška, R. (2000). Fuzzy clustering algorithms with application to rule extraction. In Szczepaniak, P.S., Lisboa, P.J.G., Kacprzyk, J.(Eds.). Fuzzy systems in medicine. Studies in fuziness and soft computing 41. Physica-Verlag HD. A Springer-Verlag company, Heidelberg, New York, 139-173.
185. Bowie, M. (2004). Fuzzy Cluster, Feature Selection, and Membership Function Optimization. Seminar Paper, University of Fribourg, Switzerland.
186. Rokach, L., Maimon, O. (2005). Clustering Methods. In : Data Mining and Knowledge Discovery Handbook. Springer US, 321-352
187. Shi, Y., Mizumoto, M. (2000). Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method. Fuzzy Sets and Systems, 112(1), 51-63.
188. Cerrada, M., Aguilar, J., Colina, E., Titli, A. (2005). Dynamical membership functions: an approach for adaptive fuzzy modelling. Fuzzy Sets and Systems, 152(3), 513–533.
189. Arslan, A., Kaya, M. (2001). Determination of Fuzzy Logic Membership Functions using Genetic Algorithms. Fuzzy Sets and Systems, 118(2), 297-306.
190. Jacob, E. (2005). A soft computing model for data clustering and application to gene grouping. PhD Thesis. Mahatma Gandhi University, Kottayam, Kerala, India.
191. Kaya, M., Alhajj, R. (2006). Utilizing Genetic Algorithms to Optimize Membership Functions for Fuzzy Weighted Association Rules Mining. Applied Intelligence 24(1), 7-15.
192. Kim, C.J., Russell, B.D. (1993). Automatic generation of membership function and fuzzy rule using inductive reasoning. In: Third International Conference on Industrial Fuzzy Control and Intelligent Systems IFIS'93, 93-96.
193. Olivas, J. (2000). Contribution to the Experimental Study of the Prediction based on Fuzzy Deformable Categories, PhD Thesis. University of Castilla-La Mancha, Spain.
194. Khan, L., Anjum, S., Bada, R. (2010). Standard fuzzy model identification using gradient methods. World Applied Sciences Journal, 8(1), 01-09.
195. Klir, G.J., Yuan, B. (2002). Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall, New Delhi.
196. Chaturvedi, D. (2008). Soft computing: techniques and its applications in electrical engineering 103. Springer: Springer Science & Business Media.
197. Derbel, I., Hachani, N., Ounelli, H. (2008, December). Membership functions generation based on density function. In: International Conference Computational Intelligence and Security, CIS'08, Suzhou, Vol. 1, 96-101.
198. Chen, S.M. (1994). A fuzzy reasoning methodology for rule-based systems. In : In Fuzzy Reasoning in Information, Decision and Control Systems. International Series on Microprocessor-Based and Intelligent Systems Engineering, 11, Vol.4.

Springer, Netherlands, 369-386.

199. Elio, R. (2002). Issues in commonsense reasoning and rationality. In: Commonsense, Reasoning and Rationality. Oxford University Press, New York, 1-52.
200. Mueller, E. (2006). Common Sense Reasoning. MIT Press.
201. Todorova, Y.M. (2006). Representing commonsense knowledge using Answer Set Programming. PhD thesis. Universidad de las Américas Puebla, Escuela de Ingeniería y Ciencias, Cholula, Puebla, México.
202. Takács, M. (2004). Critical Analysis of Various Known Methods for Approximate Reasoning in Fuzzy Logic Control. In: Proceedings of 5th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest - Hungary, November 1-7 2004.
203. Schwartz, D.G., Klir, G.J., Lewis, H.D., Ezawa, Y. (1994). Applications of Fuzzy Sets and Approximate Reasoning. Proceedings of the IEEE, 82(4), 482-498.
204. Zade, L.A. (1987). The concept of linguistic variable and its application to approximate reasoning I,II. In: Yager, R.R., Ovchinikov, S., Tong, R.M., Nguyen, H.T. (Eds.). Fuzzy Sets and Applications: Selected Papers by L.A. Zade. John Wiley&Sons, New York, 219-327.
205. Zadeh, L., Kacprzyk, J., (1992). Fuzzy Logic for the Management of Uncertainty. Wiley&Sons, New York.
206. Reiter, R. (1980). A logic for default reasoning. Artificial Intelligence, 13(1-2), 81-132.
207. McCarthy, J. (1980). Circumscription – a form of non-monotonic reasoning. Artificial Intelligence, 13(1-2), 27-39, 171-172.
208. Pelletier, F.J. (1997). What should default reasoning be, by default. Computational Intelligence, 13(2), 165-187.
209. Carbonell, J.G. (1985). Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. Technical Report CMU-CS-85-115, Carnegie Mellon University,School of Computer Science, Computer Science Department. Accssesible: http://repository.cmu.edu/cgi/viewcontent.cgi?article=2533&context=compsci
210. Pal, S.K., Shiu, S.C.K. (2004). Foundations of Soft Case-Based Reasoning. John Wiley & Sons.
211. Anthony, M., Ratsaby, J. (2013). Large margin case-based.reasoning. RUTCOR Research Repor t RRR 2-2013, Rutgers Center for Operations Research, Piscataway, NJ.
212. Hüllermeier, E. (2007). Case-Based Approximate Reasoning Theory and Decision Library B, Vol. 44. Springer Netherlands.
213. Dutta, S., Bonissone, P.P. (1991). Integrating Case Based and Rule Based Reasoning: The Possibilistic Connection. In Bonissone, P.P., Henrion, M., Kanal, L.N., Lemmer, J.H. (Eds.). Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI '90), July 27-29, 1990, MIT, Cambridge. Elswvier, Cambridge, MA, USA, 281-298.
214. Dutta, S., Bonissone, P.P. (1993). Integrating Case Based and Rule Based Reasoning. International Journal of Approximate Reasoning, 8(3), 163-203.
215. Ruspini, E.H. (1990). Possibility as similarity: The semantics of fuzzy logic. In: Proceedings 1990 AAAI Uncertainty Workshop. Cambridge, MA, Cambridge 281-

289.

216. Zadeh, L. (1972). Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges. Journal of Cybernetics, 2(3), 4-34.
217. Lee, Ch.Ch. (1990). Fuzzy logic in control systems: fuzzy logic controler, Part 1. IEEE Transactions on Systems, Man and Cybernetics, 20(2), 404-418.
218. Močkoř, J. (2010). Constructions of fuzzy logic models in categories of sets with similarities International Journal of General Systems, 39(3), 217-233.
219. Zadeh, L.A. (1979). A theory of approximate reasoning. In Hayes, J., Michie, D., Mikulich, L.I.(Eds.). Machine Intelligence 9. Halstead Press, New York, 149-194.
220. Mizumoto, M., Zimmerman, H.J. (1982). Comparison of Fuzzy Reasoning Methods. Fuzzy Sets and Systems, 8(3), 253-283.
221. de Baets, B., Kerre, E.E. (1993). The generalized modus ponens and the triangular fuzzy data model. Fuzzy Sets and Systems, 59(3), 305-317.
222. Fuller, R. (2010). A Short Survey of Fuzzy Reasoning Methods. Tutorial (slides), Abo Akademi University, Institute for Advanced Management Systems Research. Avalaible:http://uni-obuda.hu/users/fuller.robert/fuzzy-reasoning.pdf
223. Fuller, R. (1999). On fuzzy reasoning schemes. In Carlsson, C. (Ed.). The State of the Art of Information Systems in 2007. TUCS General Publications, No. 16. Turku Centre for Computer Science, Abo, 85-112.
224. Jacques, M.A.P., Pursula, M., Niittymäki, J., Kosonen, I. (2002). The impact of different approximate reasoning methods on fuzzy signal controllers. In : Proceedings of the 13th Mini-EURO Conference (Handling Uncertainty in the Analysis of Traffic and Transportation Systems) and the 9th Mtg. EURO Working Group on Transportation (Intermodality, Sustainability and Intelligent Transportation Systems)., Bari, 184–192.
225. Dubois, D., Pradel, H. (1999). Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions. Fuzzy Sets and Systems 100(Supplement), 73-132.
226. Holm, P.Karlgren, K. (1995). Theories of meaning and different perspectives on information systems. In Falkenberg, E.D., Hesse, W. (Eds.). Proceedings of the IFIP Working International Conference on Information System Concepts: Towards a Consolidation of Views (ISCO 3), Marburg, Germany, 28-30 March 1995. Chapman & Hall, 20-32.
227. Van de Cruys, T. (2010). Meaning for Meaning: The Extraction of Lexico-semantic Knowledge from Text. Proefschrit. Rijksuniversitet Groningen, Groningen .
228. Ross, D.T., Schoman, K.E. (1977). Structured Analysis for Requirements Definition. IEEE Transactions on Software Engineering, 1(3), 363-386.
229. Leite, J.C.S.P. (1989). Viewpoint Analysis: A Case Study. In: IWSSD '89 Proceedings of the 5th international workshop on Software specification and design, pp.111-119.
230. Bean, J. (2011). Data Services: Resolving the Integration Challenge with SOA, White Paper. In: Dataversity, Webshire Conferences, April 1 2011. Available at: http://www.dataversity.net/data-services-challenge-resolving-the-integration-challenge-with-soa/
231. Pathak, J., Koul, N., Caragea, D., Honavar, V.G. (2005). Discovering Web

Services over the Semantic Web. In: In Proceedings 7th ACM International Workshop on Web Information and Data Management, Bremen, Germany, 1-10.

232. Cisco (2008). Enterprise QoS Solution Reference Network Design Guide, version 4.0. Cisco Systems, Inc.
233. Anonymous (2010). A Short Fuzzy Logic Tutorial, 1-6.
234. Ragin, C.C. (2000). Fuzzy-Set Social Science. University of Chicago Press, Chicago.
235. Chen, S.J., Hwang, C.L., Hwang, F.P. (1992). Fuzzy multiple attribute decision making: methods and applications. Lecture Notes in Economics and Mathematical Systems, Vol. 375.
236. Sule, D.R. (2001). Logistics of Facility Location and Allocation. Marcel Dekker, New York, Basel.
237. Lupeikienė, A., Miliauskaitė, J., Čaplinskas, A. (2013). A Model of View-Based Enterprise Business Service Quality Evaluation Framework. Informatica, 24(4), 543–560.
238. Rao, A. (2010). Data Structures and Algorithms Using C+. Pearson Education India.
239. Karray, F., De Silva, C. (2004). Soft computing and intelligent systems design: theory, tools, and applications. Pearson Education.
240. Gao, L.: The fuzzy arithmetic mean. Fuzzy Sets and Systems., 335-348 (1999)
241. Kolmogorov, A. (1930). Sur la notion de moyenne. Cl. Sci. Mat. Mnatur. Sez. 12, Atti delle Reale Accademia Nazionale dei Lincei Mem, 323-343, (in French).
242. Anagnostopoulos, K., Doukas, H., Psarras, J. (2008). A linguistic multicriteria analysis system combining fuzzy sets theory, ideal and anti-ideal points for location site selection. Expert Systems with Applications, 35(4), 2041-2048.
243. Fu, G. (2008). A fuzzy optimization method for multicriteria decision making: An application to reservoir flood control operation, 145-149.
244. Liu, F., Mendel, J. (2008). Aggregation using the fuzzy weighted average as computed by the Karnik–Mendel algorithms. IEEE Transitions on Fuzzy Systems, 16(1), 1-12.
245. Croarkin, C., Tobias, P. (2002). Engineering statistics handbook. National Institute of Standards and Technology Information Technology Laboratory.
246. Sanchez-Pi, N., Martí, L., Molina, J., Garcia, A. (2014). High-Level Information Fusion for Risk and Accidents Prevention in Pervasive Oil Industry Environments. Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection, Springer International Publishing, 202-213.
247. Zuffi, S., Brambilla, C., Eschbach, R., Rizzi, A. (2010). Comparing image preference in controlled and uncontrolled viewing conditions. Journal of Electronic Imaging 19(4).
248. Watt, J., Van den Berg, S. (1995). Research methods for communication science. Allyn & Bacon.

APPENDICES

Appendix A. Bibliographic Research

A1. Research Questions

In accordance with the task of this research, the following research questions are formulated:

- Q1: What different understandings of the quality of service (QoS) are described in scientific publications?
- Q2: Whether, to which extent, and in which way different understandings of QoS can be generalized and integrated at higher abstraction levels?

Question Q1 must be answered by searching digital libraries and other accessible Internet sources, and extracting information from primary studies, collected by this search.

Question Q2 must be answered by analysing, generalising, aggregating, and integrating the collected information.

As a result of analysis, refinement and decomposition of question Q1, the following keywords hierarchy (Table 8) was established:

Table 8. Hierarchy of keywords

- Enterprise system
  - SOA or service-oriented system or service computing
    - Service quality
      - Business service quality or Business process quality or Quality aspects of business process
      - Quality of service
        - Web service quality or QWS
          - Quality requirements OR nonfunctional requirements OR extrafunctional requirements
            - Parameters or attributes or characteristics
              - metrics
          - QoS ontology or QoS model or QoS taxonomy
          - QoS metamodel
          - SOA QoS layer
        - QoS aware service
        - QoS aware architecture
        - Service level agreement

This hierarchy includes major terms from the research area, their acronyms, synonyms, and related terms. Synonyms are incorporated using the Boolean operator OR. The Boolean operator AND is used to link major terms.

For each concrete digital library, the hierarchy of keywords was reformulated as a query in the query language of this library, for example: “Enterprise system” + (SOA or “service oriented system” or “service computing”) + …

Boolean search string should be modified in order to adapt it to the requirements of each digital library.

A2. Review Protocol

In this section, we briefly describe the review protocol elaborated as part of the planning phase of this study. The review protocol was developed and executed according to the guidelines and hints provided by [Kitchenham and Charters, 2007], [Biolchiniet al., 2005] and previous experiences at IME-USP5 [Steinmacher et al., 2010]. The structure of the protocol is adapted from [Dyba and Dingsøyr, 2008].

Question Formularization
Question Focus: Collecting, analysis, generalisation, aggregation and integration different understandings of quality of service
Question Quality and Amplitude
Problem:
Questions: *Q1*: “What are different understandings of the term QoS” in the context of SOA? *Q2*: Is it possible to generalize the different understandings of this term by integration of different views?
Keywords and Synonyms: Notion of quality for services; services quality; QoS; web service QoS; QoS aspects for Web services; QoS ontology; QoS-aware services; QoS requirements; QoS parameters; QWS parameters; QoS attributes; QoS constraints; QoS model; QoS modelling; Quality model for Service-Oriented Systems; services in SOA; SOA QoS layer; QoS aware SOA; QoS Model for Service-Oriented Systems; non-functional characteristics of services; extrafunctional characteristics of services; web service metrics; QoS aspects for business process; QoS aware architecture; QoS metamodel; conceptual model of service quality.
Intervention: Evaluation of different understandings of quality of service.
Control: Checking approach and ad hoc reading.
Effect: Description of different understandings of QoS (different views); visualisation of statistics by diagrams, *view integration***.**

Question Formularization
Outcome measure: number of identified studies: number of selected studies in each source per year; number of selected studies of all sources per year.
Field/Scope/Confines: Publications regarding QoS understandings in the context of SOA.
Application: researchers of Software Engineering
Experimental Design: none statistical method is going to be applied.

Sources Selection
Sources Selection Criteria Definition: broad coverage in software engineering area, availability to consult articles on the web, full text availability, boolean operators support for query construction, result export capability, and academic perceived quality of content.
Studies Languages: English.
Sources Identification **Sources Search Methods**: Research through web search engines; **Search string**: "Enterprise system" AND (SOA OR "service-oriented system" OR "service computing") AND "Service quality" AND ("Business service quality" OR "Business process quality" OR "Quality aspects of business process") AND "Quality of service" AND ("Web service quality" OR QWS) AND ("Quality requirements" OR "nonfunctional requirements" OR "extrafunctional requirements") AND (Parameters OR attributes OR characteristics) AND metrics AND ("QoS ontology" OR "QoS model" OR "QoS taxonomy") AND "QoS metamodel" AND "SOA QoS layer" AND "QoS aware service" AND "QoS aware architecture" AND "Service level agreement"
Sources list: IEEE Xplore, http://ieeexplore.ieee.org ACM Digital Library, http://portal.acm.org/dl.cfm?coll=porta DBLP, http://www.informatik.uni-trier.de/~ley/db/ SciVerse ScienceDirect, http://www.sciencedirect.com/science
Sources evaluation: IEEE Xplore, ScienceDirect, DBLP fit all established criteria and will be selected. ACM Digital Library presents difficulties in exporting the results. This database is selected because of its popularity, quality and the amount of content available in this database.
Sources Selection after Evaluation: All listed sources had satisfied the sources selection criteria.
References Checking: All sources were approved.

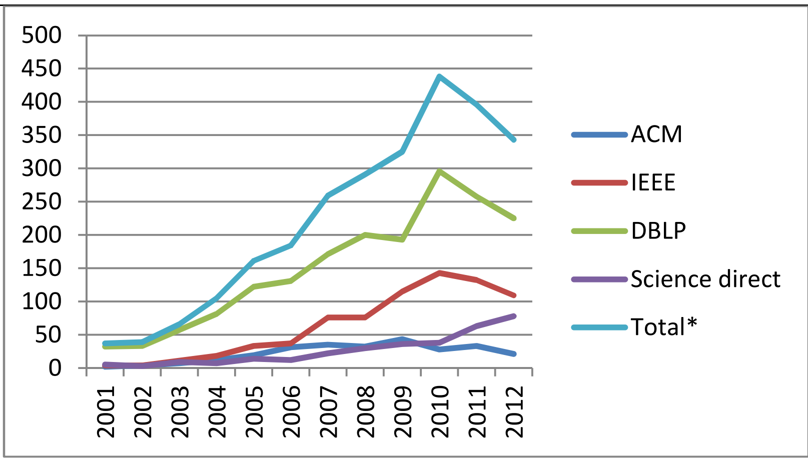
Studies Selection
Studies definition
Studies Inclusion and Exclusion Criteria Definition: **Inclusion criteria**: IC1: Universally accepted relevant fundamental works of any date published on quality of service, web service, and SOA service research and engineering related sources. IC2: Other relevant works published after 2005. IC3: Papers must be available to download. **Exclusion criteria**: EC1: Sources on quality of service, web service, and SOA service research and engineering that does not define or analyse the understanding of QoS. EC2: Relevant sources that repeat ideas described in earlier works.

Question Formularization
Studies Type Definition: The following kinds of studies related to the research topic will be selected: Journal publications (research papers, position papers, surveys, and reviews), proceedings, PhD thesis, Technical Reports; open standards.
Citation management: To assist the process of gathering and managing all the result from search engines, we used two citation management tools, namely Mendeley .JabRef (http://www.mendeley.com/) and EndNote (http://endnote.com/).   **Procedures for Studies Selection:**   • *The search strings must be run at the selected sources. To select an initial set of studies, the title, abstract and keywords of all obtained studies from web search engines are read and evaluated according to inclusion and exclusion criteria. To refine this initial set of studies, their full text is read.*   • *Search strings will be built according to the specific syntax of each selected source (see sections 2.4). Results from all sources will then be grouped in a single spreadsheet.* Duplicated and invalid results will be excluded.   • *To select an initial set of studies, the title and keywords of all obtained studies from web search engines are read and evaluated according to inclusion and exclusion criteria. All clearly irrelevant results will be discarded, i.e. papers that do not address any aspect of the research questions.*   • *The abstract of every preselected work from the previous stage will be read and another new selection will be made, based on inclusion/exclusion criteria. If reading the abstract is not sufficient to clearly understand the objectives or the problems being addressed, the review authors will also download the full article and check the study conclusions.*   • *In case multiple versions of a study exist, only the most complete version will be included.*   • *Finally, the selected studies will be fully read and write a structured abstract (executive summary) of the study.*
Selection execution
Initial Studies Selection: The complete studies list includes 3142 positions.   **Studies Quality Evaluation:** The Toulmin Model of Argument.
Information Extraction
Information Inclusion and Exclusion Criteria Definition: The extracted information from studies must contain definition or analysing of the understanding of QoS.
Synthesis of findings: The information extracted from the studies was tabulated and plotted to present basic information about the research process. The studies were cohesively grouped into categories (selected, extracted, and rejected). Sensitivity analysis was applied.

A3. Search Results

Final Comments

Number of Studies: Studies found: 3142; Rejected studies: 498; Studies extracted: 37.



Study selection analysis: We identify 3142 primary studies (see Table 9). The results of each stages of study selection process describe the stage of refinement of the study selection:

Table 9 Result of each stage of study selection process

Stage 1 Results	Total results obtained from the query submission to the source
Stage 2 Results	Results excluding irrelevant papers
Stage 3 Results	Results excluding duplicated and invalid papers
Stage 4 Results	Results applying inclusion/exclusion criteria to the papers title, keywords
Stage 5 Results	Result applying inclusion/exclusion criteria to the papers abstract and conclusion

Table 10: Number of papers in each source for each selection stage

Sources	Stage 1 Results	Stage 2 Results	Stage 3 Results	Stage 4 Results	Stage 5 Results	% of Selected results
IEEE Explore	758	689	544	544	15	40
ACM Digital Library	267	267	246	246	7	19
DBLP	1800	1800	1559	1559	4	11
Science Direct	317	317	295	295	11	30
References	3142	3073	2684	2644	37	100

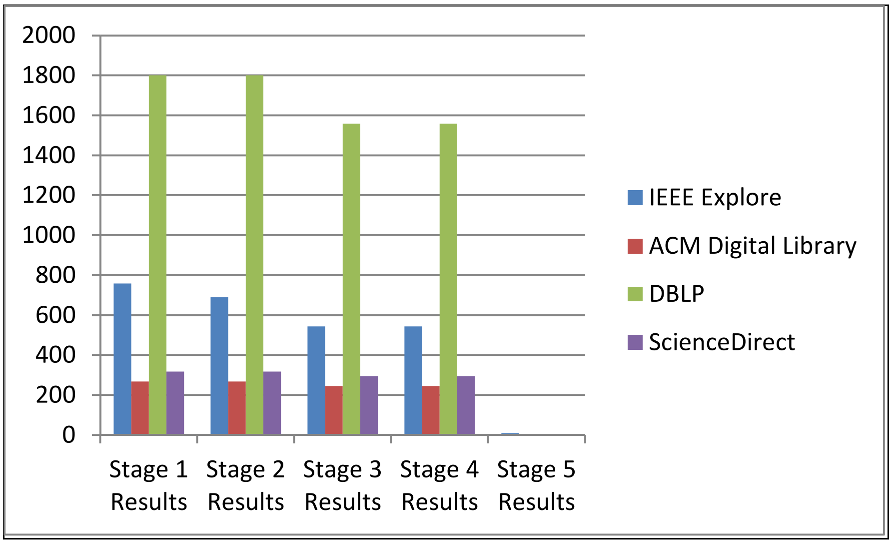


Fig. 28 Quantity of studies versus selection stages

- **Overview of primary studies:** we provide a brief overview of the primary studies of understanding of QoS by selection.
- Table 11 shows basic information on the primary studies. In addition to the all selected studies, we append open standard [OASIS, 2012] and [Cisco, 2012].

Table 11: Overview of the selected primary study

#Id	Title	Year	Paper Type
[Alben, 1996]	Quality of Experience: Defining the criteria for effective interaction design.	1996	Journal Article
[Contreras and Sourrouille, 2001]	A framework for QoS management	2001	Conference paper
[Menasce, 2002]	QoS issues in Web services	2002	Journal Article
[Box et al., 2003]	Web Services Policy Framework (WSPolicy)	2003	Open source
[Catania et al., 2003]	Web Services Management Framework	2003	Open source
[Ludwig et al., 2003]	A Service Level Agreement Language for Dynamic Electronic Services	2003	Journal Article
[Ludwig et al., 2003]	Web service level agreement (WSLA) language specification	2003	Journal Article
[Ludwig et al., 2003]	Web services qos: External slas and internal policies or: How do we deliver what we promise?	2003	Proceedings paper
[Siller and Woods, 2003]	Improving Quality of Experience for Multimedia Services by QoS Arbitration on a QoE Framework	2003	Proceedings paper
[Tosic et al., 2003]	WSOL – A Language for the Formal Specification of Classes of Service for Web Services	2003	Proceedings paper
[Bianchini et al., 2004]	QoS in ontology-based service classification and discovery	2004	Proceedings paper

#Id	Title	Year	Paper Type
[Perino et al., 2005]	A Look on Engineering Non-Functional Properties in Service Oriented Architectures	2011	Book Article
[Seth et al., 2005]	Service quality models: a review	2005	Journal Article
[Abramowicz et al., 2006]	Duality in Web services reliability	2006	Conference Paper
[Enquist et al., 2007]	Values-based service quality for sustainable business	2007	Journal article
[Kim et al., 2007]	"MOQ: Web services ontologies for QoS and general quality evaluations	2007	Journal Article
[Zapater et al., 2007]	A Proposed Approach for Quality of Experience Assurance for IPTV	2007	Proceedings paper
[Abramowicz et al., 2008]	Square based web services quality model	2008	Proceedings paper
[Aguiar et al.,]	Trends and Challenges for Quality of Service and Quality of Experience for Wireless Mesh Networks	2011	Book Article
[Du et al., 2008]	Modeling service quality for dynamic QoS publishing. In Services Computing	2008	Conference Paper
[Galster and Bucherer, 2008]	A taxonomy for identifying and specifying non-functional requirements in service-oriented development	2008	Proceedings paper
[Iancu et al., 2008]	End-to-End QoS Frameworks for Heterogeneous Networks-A Survey	2008	Conference Paper
[Qixing et al., 2008]	Modeling Service Quality for Dynamic QoS Publishing. Paper presented at the Services Computing	2008	Conference Paper
[Shekhovtsov et al., 2008]	Capturing the Semantics of Quality Requirements into an Intermediate Predesign Model	2008	Proceedings paper
[Hilari, 2009]	Quality of Service (QoS) in SOA Systems.A Systematic Review	2009	Master Thesis
[Huiyuan et al., 2009]	QoS Analysis for Web Service Composition	2009	Conference Paper
[Mabrouk et al., 2009]	A semantic end-to-end QoS model for dynamic service oriented environments	2009	Workshop paper
[Petrova-Antonova an Ilieva, 2009]	Towards a Unifying View of QoS-Enhanced Web Service Description and Discovery Approaches	2009	Journal Article
[Hernandez et al., 2010]	Evaluation Framework for Quality of Service in Web Services: implementation in a pervasive environment	2010	Master thesis
[Mintauckis, 2010]	Empirical studies of Quality of Experience (QoE) – A Systematic Literature Survey	2010	Master Thesis
[Papazoglou, et al., 2010]	Service Research Challenges	2010	Journal Article
[Goeb and Lochmann, 2011]	A software quality model for SOA	2011	Proceedings paper
[Shekhovtsov,	The Evolution of Conceptual Modelling	2011	Book Article

#Id	Title	Year	Paper Type
2011]	- From a Historical Perspective towards the Future of Conceptual Modelling		
[Szydło and Zieliński, 2011]	Model Driven Adaptive Quality Control in Service Oriented Architectures	2011	Journal Article
[Ullah, 2012]	On the ambiguity of Quality of Service and Quality of Experience requirements for eHealth services	2012	Proceedings paper
[OASIS, 2012]	Quality Model for Web Services (WSQM -2.0)	2005	Open source
[Cisco, 2012]	Enterprise Medianet Quality of Service Design 4.0—Overview	2012	Open source

Table 12 Selected paper types in percent

Paper Type	Quantity	Percent, %
Journal Article	10	27
Proceedings paper	10	27
Open source	4	11
Master thesis	3	8
Conference Paper	6	16
Book Article	3	8
Workshop paper	1	3

In total 3142 publications were selected. The papers were then further analysed for duplications, 498 duplicated papers were found. So, the final sample was 37 papers (primary sources).

Jolanta Miliauskaitė

A FUZZY INFERENCE-BASED APPROACH TO
PLANNING QUALITY OF ENTERPRISE BUSINESS SERVICES

Doctoral Dissertation

Technological Sciences, Informatics Engineering (07 T)

Editor Zuzana Šiušaitė