JOVITA BANKAUSKAITĖ

UNIFIED ARCHITECTURE FRAMEWORK-BASED TRADE STUDY METHOD FOR SYSTEM ARCHITECTURES IN MODEL-BASED SYSTEM OF SYSTEMS ENGINEERING

DOCTORAL DISSERTATION

Kaunas
2023

KAUNAS UNIVERSITY OF TECHNOLOGY

JOVITA BANKAUSKAITĖ

UNIFIED ARCHITECTURE FRAMEWORK-BASED TRADE STUDY METHOD FOR SYSTEM ARCHITECTURES IN MODEL-BASED SYSTEM OF SYSTEMS ENGINEERING

Doctoral Dissertation
Technological Sciences, Informatics Engineering (T 007)

2023, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Informatics, Department of Information Systems during the period of 2018–2023. The research was funded by the State Studies Foundation.

The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

Scientific Supervisor:
Prof. dr. Rimantas BUTLERIS (Kaunas University of Technology, Technological Sciences, Informatics Engineering – T 007).

Edited by: English language editor Dr. Armandas Rumšas (Publishing House *Technologija*), Lithuanian language editor Rozita Znamenskaitė (Publishing House *Technologija*)

Dissertation Defense Board of Informatics Engineering Science Field:
Prof. Dr. Tomas BLAŽAUSKAS (Kaunas University of Technology, Technological Sciences, Informatics Engineering – T 007) – **chairperson**;
Prof. Hab. Dr. Gintautas DZEMYDA (Vilnius University, Technological Sciences, Informatics Engineering – T 007);
Prof. Dr. Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – T 007);
Prof. Dr. Diana KALIBATIENĖ (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – T 007);
Prof. Dr. Raimundas MATULEVIČIUS (Tartu University, Estonia, Technological Sciences, Informatics Engineering – T 007).

The public defense of the dissertation will be held at 10 a.m. on 5 April, 2023 at the public meeting of Dissertation Defense Board of Informatics Engineering Science Field in A288 Meeting room at Kaunas University of Technology in “Santakos” valley.
Address: K. Baršausko 73-A228, Kaunas, LT-51383, Lithuania
Phone: (+370) 37 30 0042; email doktorantura@ktu.lt

The doctoral dissertation was sent out on the 3 of March, 2023.
The doctoral dissertation is available on the internet at http://ktu.edu and at the libraries of Kaunas University of Technology (Donelaičio 20, Kaunas, LT-44239, Lithuania) and Vilnius Gediminas University of Technology (Saulėtekio al. 14, Vilnius, LT-10223, Lithuania).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

JOVITA BANKAUSKAITĖ

ALTERNATYVIŲ SISTEMŲ ARCHITEKTŪRŲ PALYGINIMO METODAS, GRĮSTAS VIENINGU ARCHITEKTŪROS KARKASU, MODELIAIS GRINDŽIAMOJE SISTEMŲ SISTEMOS INŽINERIJOJE

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

2023, Kaunas

Disertacija rengta 2018–2023 metais Kauno technologijos universiteto Informatikos fakultete Informacijos sistemų katedroje. Mokslinius tyrimus rėmė Valstybinis studijų fondas.
Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu.

Mokslinis vadovas:
prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – T 007).

Redagavo: anglų kalbos redaktorius dr. Armandas Rumšas (leidykla „Technologija“), lietuvių kalbos redaktorė Rozita Znamenskaitė (leidykla „Technologija“)

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:
prof. dr. Tomas BLAŽAUSKAS (Kauno technologijos universitetas, informatikos inžinerija – T 007) – **pirmininkas**;
prof. habil. dr. Gintautas DZEMYDA (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T007),
prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – T 007),
prof. dr. Diana KALIBATIENĖ (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – T 007),
prof. dr. Raimundas MATULEVIČIUS (Tartu universitetas, Estija, technologijos mokslai, informatikos inžinerija – T 007).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2023 m. balandžio 5 d. 10 val. Kauno technologijos universiteto „Santakos“ slėnyje, Posėdžių kambaryje A288.

Adresas: K. Baršausko g. 59-A228, Kaunas, LT-51383, Lietuva
Tel. (+370) 37 30 0042; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2023 m. kovo 3 d.
Su disertacija galima susipažinti interneto svetainėje http://ktu.edu bei Kauno technologijos universiteto (K. Donelaičio g. 20, Kaunas, LT-44239, Lietuva) ir Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, Vilnius, LT-10223 Lietuva) bibliotekose.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to Professor R. Butleris, my research supervisor, for his countless advice and systematic assistance. I would also like to thank A. Morkevičius for his positive criticism of this research work.

Finally, I wish to thank my family for their support and encouragement throughout my studies.

TABLE OF CONTENTS

TABLE OF TABLES

LIST OF FIGURES

TERMS AND ABBREVIATIONS

Term/Abbreviation	Description
ADF	Architecture Description Framework
AHP	Analytic Hierarchy Process
Base Check	Initial evaluation of the submitted SoS architecture solutions to check whether architecture solutions meet key requirements and to determine quality
CASE tool	Computer-Aided Software Engineering tool
CBA	Call Behavior Action
Cyber-physical system	System that is built and depend on the seamless integration of computational algorithms and physical components
Deep Check	Comprehensive evaluation of plausible SoS solution architectures to choose the most preferred solution architecture, which meets the major requirements and characteristics
DMM	Domain Meta Model
DoD	Department of Defense
DoDAF	DoD Architecture Framework
FAST	Fourier Amplitude Sensitivity Analysis
FEAF	Federal Enterprise Architecture Framework
fUML	Semantics of a Foundational Subset for Executable Unified Modeling Language (UML) Models
GEOSS	Global Earth Observation System of Systems
GS	Google Schooler
Horizontal Quality Assessment	Horizontal quality assessment is designed to verify the submitted SoS solution architectures. This assessment checks only Rs domain models and its traceability links
INCOSE	International Council on Systems Engineering
JCA	Joint Capability Areas
JPSS	NASA's Joint Polar Satellite Systems
MAUT	Multi Attribute Utility Theory
MBSE	Model-based System Engineering
MCDA	Multicriteria Decision Analysis
MODAF	Ministry of Defense Architecture Framework
MOE	Measures of effectiveness
MPSA	Multiparametric Sensitivity Analysis
MSOSA	Catia Magic Systems of Systems Architect
MVA	Multivariate analysis
NAF	NATO Architecture Framework
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NESDIS	National Environmental Satellite, Data and Information Service
OAT	One-at-a-time sensitivity method
OCL	Object Constraint Language
OMG	Object Management Group
OP	Operational domain view of UAF
OSD	Office of the Secretary of Defense

Term/Abbreviation	Description
Plausible alternative	The alternative architecture solution that is selected after the Base Check
PRCC	Partial Rank Correlation Coefficient
Preferred alternative	Alternative architecture solution that is the most balanced solution
Profile	Extension mechanism to UML language.
PSCS	Precise Semantics of UML Composite Structure
PSOT	Precise Semantics of Time
PSSM	Precise Semantics of UML State Machines
RFP	Request For Proposal
RM	Requirements Management
RS	Resources domain view of UAF
SA	Sensitivity Analysis
Sanity Check	Engineering judgement to ensure that the proper decision is made.
SE	System Engineering
SEBoK	SE Body of Knowledge
Sensitivity Analysis	Method that studies the effects of input uncertainty on the response to the model output
SoI	System of Interest
SoS	System of Systems
SoSE	System of Systems Engineering
ST	Strategic domain view of UAF
SysML	Systems Modeling Language
TOGAF	The Open Group Architectural Framework
TOPSIS	Technique for Order Preferences by Similarity to Ideal Solutions
Trade Study	"Decision-making activity used to identify the most acceptable technical solution among a set of proposed solutions" [1]
Trade Study Quality	Reducing potential errors in evaluating alternative SoS architectures. It follows reducing the risk of choosing an inappropriate alternative SoS architecture
UAF	Unified Architecture Framework
UAFP	UAF Profile
UT3SA	UAF-based Trade Study method for SoS Architectures
VCSAF	Vice Chief of Staff of the Air Force
Vertical Quality Assessment	Vertical quality assessment is designed to verify the submitted SoS solution architectures against the requirements model. This assessment checks St, Op, and Rs domain models and its traceability links

INTRODUCTION

Motivation

The adoption of model-based systems engineering (MBSE) continues to grow in the industry, as does the interest in MBSE as a research topic among systems engineering researching universities [2]. Also, MBSE remains a leading practice in the International Council on Systems Engineering (INCOSE) vision for 2035 [3]. The main factor for adopting MBSE in the industry is the engineer's ability to develop systems with traceability to requirements using a single integrated architecture model that allows all types of automated analysis, such as impact analysis, gap analysis, trade study, and various simulations.

The application of MBSE has become the key to solving complex real-world problems [4] [5] [6]. Complexity issues are particularly prevalent when constituent systems, which are independent and operable, must communicate together to achieve a specific higher goal. The systems engineer focuses primarily on the design of dedicated domain-specific systems. However, it is now widely recognized that systems and devices are no longer stand-alone, but, instead, are interconnected as part of a broader system of systems (SoS). From the systems engineering point of view, this is the SoS level. At this level, one of the main concerns is architecture evaluation and trade study analysis, which may lead to different criteria and methods for identifying and comparing alternatives to keep the architecture in line with the assigned budgets and applicable deadlines.

Trade studies have become an indispensable engineering activity and are often used in the early stages of system architecture development. A trade study evaluates alternatives based on criteria and systematic analysis so that to select the most balanced alternative to attain the desired objectives [7] [1]. In the cyber-physical systems or SoS design, a trade study is often a highly complex, time-consuming, and expensive activity requiring extensive resources and broad knowledge [8]. Conducting a trade study can take from several months to more than two years, whereas the cost may range from several hundred thousand to several million dollars, depending on its scope and complexity [9]. For example, the Vice Chief of Staff of the Air Force (VCSAF) study revealed that the average cost of a trade study is $15 million, and the average duration of order-to-delivery is 21 months [9]. However, despite the high costs, the trade study is considered a valuable investment, as inappropriate decisions may lead to higher material losses in the later stages of system development.

Typically, architectures are individually evaluated on each parameter of interest, and the results are manually compared [10]. The most common shortcomings in a trade study are the lack of transparency in the final decision and the over-emphasis on the lowest price criterion [11]. Also, the supplement of new alternatives at mid-term evaluation poses a number of challenges and significantly increases costs compared to selecting the same range of alternatives at the beginning of the evaluation.

Nowadays, there are several architecture description frameworks (ADF) and modeling languages that are extensively used when it comes to the SoS, such as the NATO Architecture Framework (NAF) [12] [13], or Department of Defense

Architecture Framework (DoDAF) [14] [15] [16]. Over time, the system of systems engineering (SoSE) continues to evolve to simplify intricate processes and concepts. A new AF, called the Unified Architecture Framework (UAF), has recently emerged, and it became the official Object Management Group (OMG) standard in 2017 [17]. UAF has been designed to support trade studies at various levels, including three core domains: *Strategic*, *Operational*, and *Resources* [17]. However, despite the mechanism provided to capture the required data, UAF has no process that guides the way to conduct trade studies. Although several processes are found in the literature that provide a step-by-step description of trade study analysis, no one has given details yet as to how trade study analysis could be automated in a MBSE along with the already existing ADF, languages, and architecture evaluation methods. Sophisticated models, advanced visualization, and highly integrated, multidisciplinary simulations would allow systems engineers to evaluate more alternative designs faster and more comprehensively.

Although trade study methods are used in the system engineering domain, so far, their applicability and adaptation to SoS architectures in the MBSE environment have not been extensively studied. In addition, as digitalization has been accelerating and the number of organizations using MBSE has thus been increasing, there is a lack of scientifically proved trade study methods that can be applied in the MBSE environment [18] [19].

Object and Scope of the Dissertation

The research object is the trade study method to evaluate alternative SoS architectures in the MBSE environment.

The scope of the research covers the following topics:

- SoSE and MBSE;
- SoS modelling approaches and frameworks;
- Alternatives evaluation methods;
- Standards and methods for automated architecture evaluation in the MBSE environment.

Goal of the Dissertation

The goal of the dissertation is to increase the quality of a trade study by providing means to evaluate alternative SoS architectures in the MBSE environment aimed at automating the evaluation of SoS architectures by using model execution.

Objectives of the Dissertation

To achieve the goal of the dissertation, the following list of objectives must be accomplished:

1. Analyze the existing ADFs, by focusing on their application for trade study with the ability to automate the evaluation of SoS architectures.
2. Analyze the currently available trade study processes and architecture evaluation methods while focusing on their practical application in the MBSE environment.

3. Develop a model-based trade study method that includes evaluation of alternative SoS solution architectures in the MBSE environment.
4. Implement the proposed method in the CASE tool.
5. Experimentally evaluate the suitability of the proposed method for several alternative SoS architectures.

Research Methodology

The research methodology of this dissertation is based on the design research process [20], which can be divided into three main parts.

First, the existing literature on system engineering (SE), MBSE, ADF, trade study processes and architecture evaluation methods have been analyzed. The revised literature has been systematically summarized by using comparison and classification methods.

Second, to develop a trade study method for SoS architectures which would be applicable in the MBSE environment. The following methods were used: conceptual modeling methods, UAF domain metamodel, synthesis, integration, and metamodeling.

Third, a case study was used to evaluate the method proposed in this thesis. The case study was modeled on the guidelines introduced and experimentally evaluated by running a trade study in the MBSE environment. The experimental evaluation included quantitative data analysis and classification methods.

Defended Statements

The defended statements of this thesis are presented below:

1. The MBSE environment and the foundational UML subset (fUML) are suitable means to automate the evaluations of alternative SoS architectures within the trade study process that allow to reduce the errors associated with this process.
2. The existing UAF domain metamodel needs to be extended with new concepts to define and perform a trade study in the MBSE environment.
3. The UT3SA method provides the required process, guidelines, evaluation algorithms, and measurements to increase the quality of a trade study by reducing errors in evaluating alternative SoS architectures.

Scientific Novelty

The scientific novelty of this thesis is presented below:

1. The proposed UAF-based trade study method for SoS architectures (UT3SA) is the only known method that encompasses all of the following:

1.1. Defined trade study process based on a formal UAF metamodel and profile. Additionally, the process includes the necessary steps when a trade study is conducted as a result of an acquisition process.

1.2. Vertical (cross-domain oriented) and horizontal (single-domain oriented) alternative SoS architecture quality assessment aspects, including method-specific quality measurement.

1.3. Evaluation of qualitative and quantitative parameters of SoS architectures.
1.4. This method is fully model based and can be used in the MBSE environment.
1.5. The proposed method defines two algorithms to run SoS architectures evaluation by using selection criteria and sensitivity analysis in an automated way in the MBSE environment. Algorithms are based on fUML principles.

2. The existing UAF domain metamodel and profile are supplemented with the necessary extensions to prepare and run a trade study for SoS architectures.
3. The proposed method of trade study is one of the first methods in the area of SoS in the MBSE environment at the time of publication.

Practical Significance

The main practical significance of this research is to increase the quality of the trade study of SoS architectures using the MBSE environment. These improvements are driven by the proposed method, which provides means to:

- Describe and relate each trade study selection criteria to the expected SoS architecture. The proposed method provides guidelines on how to define the trade study in the MBSE environment using the UAF domain metamodel. Linking the selection criteria of the trade study to the expected SoS architecture increases the accuracy of the trade study.
- Identify and remove poor quality SoS architectures before running a thorough evaluation of each alternative SoS architecture. The quality evaluation of alternative SoS architecture solutions increases the quality of the study, as the final results of the evaluated alternative may be distorted due to the poor quality of the architecture.
- Determine the level of compliance of alternative SoS solution architectures according to the established selection criteria in the MBSE environment. The automated evaluation method reduces the probability of occurring errors and enables a simpler way to change the set of alternatives compared to the traditional methods.
- Identify alternative SoS architectures to be checked in the sensitivity analysis. Sensitivity analysis is performed with the objective to determine which alternative has a greater advantage when the level of compliance is very similar. Incorporating sensitivity analysis into a trade study improves the quality of the study itself by introducing an additional evaluation step to verify the final decision.
- Determine the sensitivity index for each selection criterion. The most sensitive criteria are selected to compare alternative SoS architectures with similar scores and identify their compliance with the selected criteria. This evaluation increases the accuracy of the choice of the most balanced alternative.

The application of the method to eight alternative SoS solution architectures has demonstrated that the proposed method can be and is actually being applied in practice. In order to use the proposed method, it is necessary to implement it in one of the SoS modeling CASE tools. In this thesis, the UAF-based UT3SA profile was implemented by using the *Catia Magic Systems* of the *Systems Architect v2021x Refresh2 CASE* tool.

The proposed method is strictly based on the principles of UAF. As UAF is based on UML/SysML, it can be assumed that the proposed method can be adapted at the SE level as well.

Scientific approval

Two articles presenting the results of the thesis have been published in a scientific journal indexed in the *Clarivate Analytics Web of Science* database. In addition, the results of the thesis have been presented at four international conferences in Hungary, the USA, South Africa, and Lithuania. The corresponding publications were published in the conference proceedings. A detailed list of publications can be found in the chapter *LIST OF SCIENTIFIC ARTICLES ON THE DISSERTATION TOPIC*.

Structure of the dissertation

The dissertation is structured as follows: Chapter 1 contains analysis of SoSE, MBSE, ADFs, trade study processes and evaluation methods for alternative architecture solutions; Chapter 2 contains the definition of a new UT3SA trade study method for the SoS architecture in the MBSE environment; Chapter 3 contains evaluation of the suitability of the proposed UT3SA method on the grounds of applying it to eight alternative SoS solution architectures. The total scope of the dissertation is 180 pages with appendixes (157 pages without appendixes); it features 64 figures and 39 tables.

All figures and tables without citations have been designed and/or compiled by the author of the dissertation.

1 SYSTEM OF SYSTEMS ARCHITECTURE MODELING AND ANALYSIS

The purpose of this chapter is to define the context of the dissertation and identify the existing problems in the trade study of SoS architectures.

This chapter consists of an SoS definition, an MBSE definition, analysis of ADFs, analysis of standard-based execution methods of SoS architecture models, and analysis of trade study processes and evaluation methods.

1.1 Definition of a system of systems engineering

The traditional SE and SoSE are interconnected, but they cover different fields of study. SoSE is defined as: “the process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability that is greater than the sum of the capabilities of the constituent parts. In addition, this process emphasizes the process of discovering, developing, and implementing standards that promote interoperability among systems developed via different sponsorship, management, and primary acquisition processes” [21]. Meanwhile, SE is defined as: “the process by which a customer’s needs are satisfied through the conceptualization, design, modeling, testing, implementation, and operation of a working system” [21].

Initially, SoSE was identified for the defense sector. However, now SoSE concepts and principles are being applied across other governmental, civil, and commercial areas, such as transportation, energy, health care, defense, rail, business, disaster response, etc. [22] [23] [24]. The *SE Body of Knowledge* (SEBoK) [22] stated that SoSE allows the SE community to have the opportunity to define software intensive systems of the 21st century. Although SE is a well-established field, SoSE is a global challenge for current system engineers. In general, SoSE requires that accounts be taken not only of factors usually related to engineering but also of sociotechnical and socioeconomic phenomena.

Table 1.1 presents the differences between the traditional SE and SoSE by summarizing sources [21] [25] [26]. Based on the analysis of the differences, it can be concluded that one of the fundamental differences is that SE addresses the development of monolithic systems. On the contrary, SoSE addresses the development of constituent systems, and the synergy of these systems is called SoS.

Table 1.1. Difference between SE and SoSE

Criteria	SE	SoSE
Purpose	Development of a single system that meets the needs of the stakeholders	Development of new SoS capacity using the synergies of legacy systems
System architecture	System architecture is established at the beginning of the life cycle and is relatively stable	Dynamic architecture reorganization as needs change; use of service architecture approaches as enablers
System interoperability	Definition and implementation of specific	Component system can be usefully operated independently of the SoS by

Criteria	SE	SoSE
	interface requirements for integration of components into the system	protocols and standards that are essential to enable the interoperability of the system
System 'ilities'	Reliability, maintainability, and availability are the typical 'ilities'	Added 'ilities', such as flexibility, adaptability, and composability
Acquisition and management	Centralization of system acquisition and management	Component systems are acquired separately and are managed independently as a system
Anticipation of needs	Activity of the conceptual phase is to determine the needs of the system	Intensive analysis of the concept phases is followed by continuous prediction and supported by continuous experimentation

SoS is composed of independent constituent systems that act jointly towards a common goal through the synergism between them. Currently, several definitions of SoS can be found in the literature, some of which depend on the application:

- *Definition 1*: "SoS brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its own management, goals, and resources while coordinating within the SoS and adapting to meet SoS goals" [27].
- *Definition 2*: "A collection of systems, each capable of independent operation, that interoperate together to achieve additional desired capabilities" [28].
- *Definition 3*: "SoS is a set or arrangement of systems that results from independent systems integrated into a larger system that delivers unique capabilities" [29].
- *Definition 4*: "SoS is the large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal. The goal may be cost, performance, robustness, etc." [30].

In the synthesis of SoS definitions, Maier presented five main characteristics [31] to determine whether a system of interest (SoI) can be considered an SoS. These five characteristics are as follows:

- Operational independence
- Managerial independence
- Geographical distribution
- Emergent behavior
- Evolutionary development processes.

Maier also highlighted that the operational and managerial independence of component systems are the main distinguishing features by which an SoI is considered to be an SoS. A system that does not meet these two characteristics is not considered an SoS, regardless of the complexity or geographical distribution of its components.

Today, there are several specialized research organizations of SoS, such as *INCOSE* [32], the *Office of the Secretary of Defense* (OSD) [33], and the *Group of Global Earth Observation System of Systems* (GEOSS) [34].

1.2 Architecture description frameworks

In order to model architectures in detail, some guidance is necessary, which is provided by ADF. Currently, several definitions of ADF can be found in the literature:

- *Definition 1*: "Conventions, principles, and practices for the description of architectures established within a specific domain of application and/or community of stakeholders" [35].
- *Definition 2*: "An architecture framework is an encapsulation of a minimum set of practices and requirements for artifacts that describe a system's architecture. Models are representations of how objects in a system fit structurally in and behave as part of the system" [36].
- *Definition 3*: "An architecture framework is a tool which can be used for developing a broad range of different architectures. It should describe a method for designing an information system in terms of a set of building blocks and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks." [37].

ADFs assist in the specification of architectures by serving as a template that organizes the structure of architecture into projections called 'views'. ADFs define a set of views that focus on a particular aspect of the architecture.

Currently, several ADFs are extensively used when it comes to SoS, such as the DoDAF, NAF, MODAF, UAF, *Federal Enterprise Architecture Framework* (FEAF), and *The Open Group Architectural Framework* (TOGAF). Each of these ADFs is briefly described in the following.

DoDAF

DoDAF is developed for the *US Department of Defense* (DoD) to provide a visual infrastructure to address specific stakeholders' concerns through different viewpoints. It helps to ensure that architectural artifacts are consistently defined and characterized in accordance with the specific requirements of the project or mission. DoDAF allows managers to make more effective critical decisions by organizing the dissemination of information across departments, *Joint Capability Areas* (JCA), mission, component, and program boundaries [38]. DoDAF focuses more on architectural data than on architectural artifacts. This framework defines the method for specifying the SoS by using the architectural terms of the DoD [39].

NAF

NAF is developed by the *North Atlantic Treaty Organization* (NATO) and comes from DoDAF. NAF's objective is to provide standards for the development and description of architecture for military and commercial purposes [40]. NAF aims to enable many organizations, including NATO and other national defense initiatives, to understand, compare, justify, and link the architecture developed in its framework.

NAF defines the methodology, viewpoints, stakeholder viewpoints, and metamodels [40].

MODAF

MODAF is developed by the British Ministry of Defense to support defense planning and change management activities. MODAF ensures accurate, complete, and consistent collection and presentation of information and helps to understand complex problems [41]. The main advantages of MODAF are the improvement of interoperability and the implementation of the system. The framework supports various MOD processes, including capacity management, acquisition, and maintenance. The MODAF architecture is designed as a parallel and coherent model that provides a comprehensive view of the enterprise. MODAF defines various relationships for the integration of architectural elements [41].

UAF

UAF is created based on the needs of UML/SysML and the military community to create standardized and consistent architectures based on the DoDAF and MODAF [42]. UAF consists of three main components: (1) framework – a collection of domains, model types, and viewpoints, (2) metamodel – a set of types, tuples, and individuals used to build views based on specific viewpoints, (3) profile – SysML-based implementation of metamodels to apply principles and best practices of system engineering based on models while building views. UAF provides a set of rules for creating consistent enterprise architectures (models) based on general enterprise concepts and rich semantics. These models become a repository for extracting various views [43].

FEAF

FEAF is developed by the federal government of the United States. It provides a common methodology to integrate strategic, business, and technological management as part of organizational design and performance improvement [44]. The government uses EAF to help develop large-scale and complex system development processes through the organization’s definition of enterprise architecture. The architectural segment is created independently, depending on the structural guidelines, and each segment is considered its federal enterprise.

TOGAF

TOGAF is based on the DoD Information Management Technology Architecture Framework [45]. TOGAF focuses on business applications that are critical to the mission and that use open system components. TOGAF provides and explains rules and develops good principles for the development of the system architecture. TOGAF contains three levels of principles: (1) it supports decision-making across the enterprise; (2) it guides IT resources; (3) it supports the principles of architecture for the development and implementation.

The following section shall present a comparison of six ADFs (DoDAF, NAF, MODAF, UAF, FEAF, and TOGAF). The ADFs are compared in order to determine the most appropriate framework for SoS based on which the trade study method will be developed.

1.2.1 Comparison of architecture description frameworks

This section provides the ADFs comparison by identifying and summarizing the main differences. The comparison analyzes the six frequently used ADFs: DoDAF v2.02, MODAF v1.2, NAF v4.0, UAF v1.1, FEAF v2.3, and TOGAF v9.2. The comparison criteria are as follows:

- **Domain** – application in defense or industry. The criterion determines the universality of the framework.
- **Viewpoints** – a set of supported viewpoints. The criterion determines which aspects of architecture the framework focuses on.
- **Metamodel/Modeling language** – support of native metamodel or standard modeling language. The criterion determines how comprehensive the framework is and what formalism level it supports.
- **Tool support** – a set of modeling tools that supports the framework. The criterion determines the possibility of using the framework in practice through a modeling tool.
- **Prevalence among researchers** – the prevalence level of the research community. The criterion determines whether the framework is being studied or developed in scientific work. The *Google Schooler* (GS) database is used to find scientific works on certain ADF by providing the number of all publications and the number of publications since 2017.

Comparison of ADFs according to the specified criteria is provided in Table 1.2 by summarizing sources [37], [38], [40], [41], [44], [46], [47].

Table 1.2. ADFs comparison

ADF/ Criteria	Domain	Viewpoints	Metamodel / Modeling language	Tool support	Prevalence among researchers
DoDAF	Defense	▪ All ▪ Capability ▪ Data and Information ▪ Operational ▪ Project ▪ Services ▪ Standards ▪ Systems	▪ IDEAS ▪ UPDM ▪ BPMN	▪ Sparx EA ▪ Catia MSoSA ▪ Unicom SA ▪ Hopex Mega ▪ PTC Integrity Modeler ▪ IBM Rhapsody	▪ GS: 5140 o Since 2017: 1560
NAF	Defense	▪ Concepts ▪ Service specification ▪ Logical specification ▪ Physical resource specification ▪ Architecture foundation	▪ Native metamodel ▪ UPDM	▪ Catia MSoSA ▪ Unicom SA ▪ Hopex Mega ▪ PTC Integrity Modeler ▪ IBM Rhapsody	▪ GS: 1040 o Since 2017: 325

ADF/ Criteria	Domain	Viewpoints	Metamodel / Modeling language	Tool support	Prevalence among researchers
MODAF	Defense	▪ All ▪ Strategic ▪ Acquisition ▪ Operational ▪ Service-oriented ▪ Systems ▪ Technical standards	▪ MODEM ▪ UPDM	▪ Sparx EA ▪ Catia MSoSA ▪ Unicom SA ▪ Hopex Mega ▪ PTC Integrity Modeler ▪ IBM Rhapsody	▪ GS: 2880 o Since 2017: 1100
UAF	Defense, Industry	▪ Architecture management ▪ Strategic ▪ Operational ▪ Services ▪ Personnel ▪ Resources ▪ Security ▪ Projects ▪ Standards ▪ Actual resources	▪ UAFP	▪ Sparx EA ▪ Catia MSoSA ▪ PTC Integrity Modeler ▪ Unicom SA ▪ IBM Rhapsody	▪ GS: 5280 o Since 2017: 2020
FEAF	Industry	▪ Strategic plans ▪ Business activities ▪ Data and information ▪ Systems and applications ▪ Networks and architecture	▪ Native metamodel	▪ Unicom SA ▪ Hopex Mega	▪ GS: 2150 o Since 2017: 805
TOGAF	Industry	▪ Preliminary ▪ Architecture vision ▪ Business architecture ▪ Data architecture ▪ Application architecture ▪ Technology architecture ▪ Opportunities and solutions ▪ Requirements managements	▪ Native metamodel ▪ ArchiMate ▪ UML	▪ Sparx EA ▪ Catia MSoSA ▪ Unicom SA ▪ Hopex Mega ▪ Troux Planview	▪ GS: 12500 o Since 2017: 5580

Based on the results of the ADFs comparison, it can be stated that only UAF can be applied in industry and defense domains; other frameworks have only one dedicated domain. Additionally, all defense frameworks have a general tendency of viewpoints, however, only UAF introduces such viewpoints as security and personnel. Industrial frameworks use very similar viewpoints but differ from defense frameworks. FEAF is the least tool-supported framework as other frameworks are supported by similar amounts of tools. The most prevalent ADFs among researchers are DoDAF, UAF, and TOGAF.

1.2.2 Unified architecture framework

In response to the demands of the industry and the evolution of the ADFs, UPDM 3.0 (later renamed as UAF) was initially developed in 2013 to meet the emerging needs of the SoS [48]. However, the UPDM 3.0 working group subsequently established a list of requirements that arise from the increasing complexity and the rising costs of the systems. As a result, such requirements as the architecture modeling support for the defense, industry, and the government organizations, security support, analysis, and other areas have been introduced [49].

In 2017, a new framework called UAF was introduced which has become the official standard of OMG [17]. UAF is a unified framework whose fundamental concepts are based on three military frameworks: DoDAF, MODAF, and NAF. Furthermore, "UAF extends the scope of UPDM and generalizes it to make it applicable to commercial as well as military architectures" [46].

The UAF has become an essential upgrade for both the DoD and commercial organizations. UAF architecture models provide an opportunity to understand the complex relationships between organizations, systems, and SoS and enable the analysis of these systems [18]. UAF as a framework and UAFP as a language have been applied to the development of various SoS cases.

Many successful uses of UAF can be found in the literature. The *National Environmental Satellite, Data and Information Service* (NESDIS) ground enterprise and NASA's *Joint Polar Satellite Systems* (JPSS) ground project are successful examples. At the end of these projects, it was noted that the MBSE approach reduces rework, increases accuracy, and allows firm and informed decision-making [50]. US-DOD SoS modeling research projects describe how UAF use removes problem-solving restrictions as per [51]. Sources [52] and [53] describe how a UAF model can be used to significantly influence the ongoing SE efforts, as well as a software architecture for application development.

Fig. 1.1 shows three main components of UAF [46] [18]:

- **framework** – a group of domains, model kinds, and viewpoints;
- **metamodel** – a group of types to construct views based on specific viewpoints;
- **profile** – SysML-based implementation of the metamodel to apply MBSE principles and best practices in developing views.

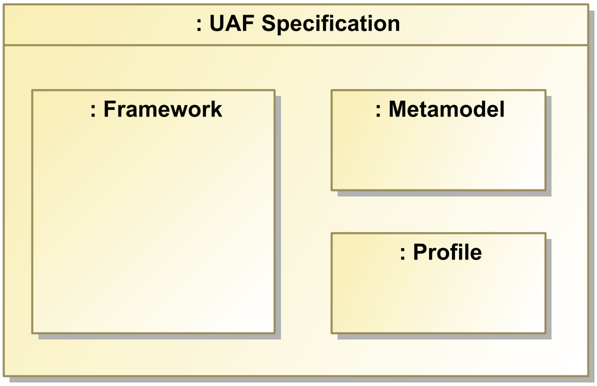


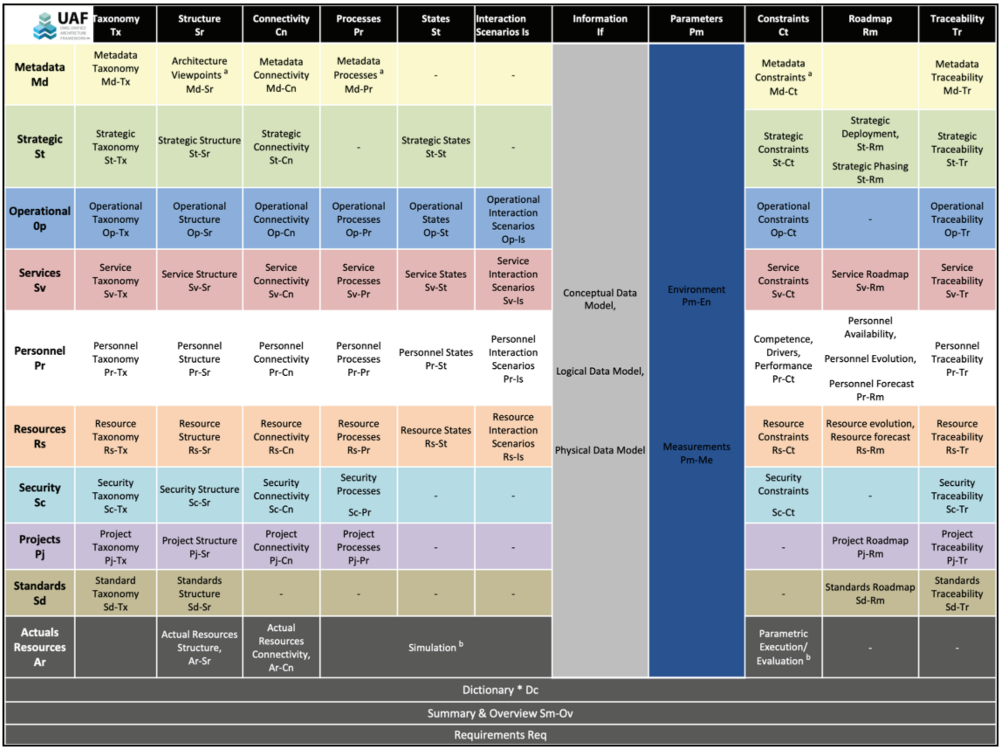
Fig. 1.1. UAF components

UAF Profile (UAFP)

UAFP defines metamodel extensions in the context of UML to adapt metamodel and organize it to a specific platform or domain. The UAFP specification reuses a subset of UML 2.5.1 and SysML 1.5 and provides additional extensions [54].

UAF Framework

UAF is formatted in a grid (Fig. 1.2) that contains rows and columns. Rows represent domains that can be used when modeling an SoS architecture. The columns represent the model kinds. The intersection of a row and a column is called a viewpoint. This grid is provided in the UAF standard as a structuring formalism to organize the 71 view specifications defined within the UAF [46]. The UAF grid is the basis for developing a domain-specific architecture choosing only the viewpoints necessary for a specific context.



UAF	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Interaction Scenarios Is	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Metadata Md	Metadata Taxonomy Md-Tx	Architecture Viewpoints [a] Md-Sr	Metadata Connectivity Md-Cn	Metadata Processes [a] Md-Pr	-	-			Metadata Constraints [a] Md-Ct		Metadata Traceability Md-Tr
Strategic St	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	-	Strategic States St-St	-			Strategic Constraints St-Ct	Strategic Deployment, St-Rm Strategic Phasing St-Rm	Strategic Traceability St-Tr
Operational Op	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Interaction Scenarios Op-Is			Operational Constraints Op-Ct	-	Operational Traceability Op-Tr
Services Sv	Service Taxonomy Sv-Tx	Service Structure Sv-Sr	Service Connectivity Sv-Cn	Service Processes Sv-Pr	Service States Sv-St	Service Interaction Scenarios Sv-Is	Conceptual Data Model,	Environment Pm-En	Service Constraints Sv-Ct	Service Roadmap Sv-Rm	Service Traceability Sv-Tr
Personnel Pr	Personnel Taxonomy Pr-Tx	Personnel Structure Pr-Sr	Personnel Connectivity Pr-Cn	Personnel Processes Pr-Pr	Personnel States Pr-St	Personnel Interaction Scenarios Pr-Is	Logical Data Model,		Competence, Drivers, Performance Pr-Ct	Personnel Availability, Personnel Evolution, Personnel Forecast Pr-Rm	Personnel Traceability Pr-Tr
Resources Rs	Resource Taxonomy Rs-Tx	Resource Structure Rs-Sr	Resource Connectivity Rs-Cn	Resource Processes Rs-Pr	Resource States Rs-St	Resource Interaction Scenarios Rs-Is	Physical Data Model	Measurements Pm-Me	Resource Constraints Rs-Ct	Resource evolution, Resource forecast Rs-Rm	Resource Traceability Rs-Tr
Security Sc	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-			Security Constraints Sc-Ct	-	Security Traceability Sc-Tr
Projects Pj	Project Taxonomy Pj-Tx	Project Structure Pj-Sr	Project Connectivity Pj-Cn	Project Processes Pj-Pr	-	-			-	Project Roadmap Pj-Rm	Project Traceability Pj-Tr
Standards Sd	Standard Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-			-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actuals Resources Ar		Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation [b]					Parametric Execution/ Evaluation [b]	-	-
Dictionary * Dc											
Summary & Overview Sm-Ov											
Requirements Req											

Fig. 1.2. UAF grid [46]

Table 1.3 describes ten UAF domains by providing their purpose, a list of responsible stakeholders, and concerns on the grounds of [54] [46] [55] [51].

Table 1.3. UAF domains

Domain	Purpose	Stakeholders	Concerns
Metadata (Md)	▪ Provide information related to the entire architecture and provide supporting information	▪ Enterprise architects ▪ Technical managers	▪ Capture metadata relevant to the SoS architecture
Strategic (St)	▪ Describe the capability taxonomy, composition, dependencies, and evolution	▪ Capability portfolio managers	▪ Capability management process
Operational (Op)	▪ Describe the requirements, operational behavior, structure, and exchanges needed to support capabilities ▪ Define all operational elements regardless of the solution	▪ Business architects ▪ Executives	▪ Illustration of the logical architecture of the enterprise or SoS
Services (Sv)	▪ Define service specifications, including the required and provided service levels of these specifications	▪ Enterprise architects ▪ Solution providers	▪ Specifications of services needed to exhibit capability

Domain	Purpose	Stakeholders	Concerns
	necessary to exhibit a capability and support an operational activity	▪ Systems engineers ▪ Software architects ▪ Business architects	
Personnel (Ps)	▪ Clarify the human roles in the development of architectures	▪ Human resources ▪ Solution providers ▪ PMs	▪ Human factors
Resources (Rs)	▪ Capture a solution architecture that includes resources such as software, artifacts, capability configurations, and natural resources that meet operational requirements	▪ Systems engineers ▪ Resource owners ▪ Implementers ▪ Solution providers ▪ IT architects	▪ Definition of solution architectures to implement operational requirements
Security (Sc)	▪ Define the security assets, restrictions, controls, and measures needed to address specific security concerns	▪ Security architects ▪ Security engineers	▪ Addresses security restrictions and information assurance attributes
Projects (Pj)	▪ Describe the projects, project milestones, how those projects deliver capabilities, and projects dependencies	▪ PMs ▪ Project portfolio managers ▪ Enterprise architects	▪ Project portfolio, projects, and project milestones
Standards (Sd)	▪ Define the technical, operational, and business standards applicable to the architecture	▪ Solution providers ▪ Systems engineers ▪ Software engineers ▪ Systems architects ▪ Business architects	▪ Technical and non-technical standards applicable to the architecture
Actual Resources (Ar)	▪ Define the actual resource configurations expected or achieved and the actual relationships between them	▪ Solution providers ▪ Systems engineers ▪ Business architects ▪ Human resources	▪ Analysis such as trade study, what-if, and V&V

The three domains of the UAF are considered fundamental; they reflect different levels of abstraction, and these domains are strategic, operational, and resources [56]. Typically, architecture development occurs from the strategic domain to the resources domain. In the development of the architecture process, the trade study can be conducted between different operational scenarios or/and different resources configurations (Fig. 1.3).

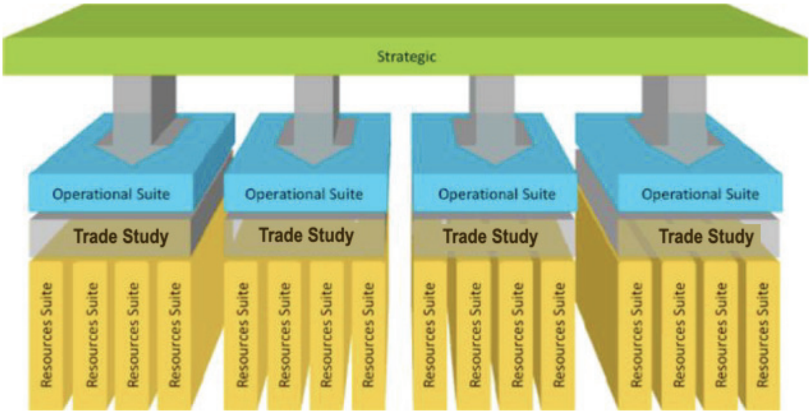


Fig. 1.3. Abstraction levels of architecture framework [57]

The UAF standard does not define how a trade study is conducted, nor does it specify a formal trade study process. UAF provides only a basis for collecting the data required for trade studies from the architecture point of view. However, there are no concepts needed to define a set of criteria for a trade study or to identify a level of priority.

1.3 Definition of model-based system engineering

For a long time, about forty years, the practice of SE was linear: (1) documentation of requirements; (2) carrying out the analysis; (3) development of a conceptual design. Advances in the lifecycle of system development have introduced new methods for designing and developing systems, such as waterfall, spiral, incremental, and sprint-based. However, the lack of integration from one stage to another results in a longer time to build a system. In addition, it increases the cost of fixing errors found at the integration points in the cycle stages.

"MBSE is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [3]. MBSE is practiced in many industries around the world. For example, the world's largest telescopes [58], fighter jets [59] and autonomous driving cars [59] have been developed with MBSE.

MBSE is often compared to the traditional document-centric engineering. The main characteristic of the documented-based approach is that a system is designed by using document artifacts, such as text documents, drawings, and spreadsheets. Meanwhile, MBSE integrates the system data into a consistent and integrated model.

The OMG organization highlights five benefits that MBSE brings to SE [60]:

- Improves accuracy, consistency, and traceability;
- Includes behavioral analysis, system architecture, requirement traceability, performance analysis, simulation, testing, and others;
- Formalizes system development practices by using models;
- Integrates data between discipline-specific engineering tools, including hardware and software design, analysis, simulation, and testing;
- Facilitates general understanding of the system within the development team.

In 2020, NASA noted that MBSE "has been increasingly embraced by both industry and government as a means to keep track of system complexity. It allows the engineer to represent the system in a comprehensive computer model allowing for better traceability, tracking, and information consistency" [61]. Also, a comprehensive study [62] has shown that the strict application of MBSE has a significant advantage over document-centric engineering in terms of the potential cost and schedule savings.

Article [63] presented the results of detailed research on the value and benefits of MBSE in the analysis of 847 articles. The most commonly cited advantages of MBSE in the literature are: (1) increased traceability; (2) improved consistency; (3) reduced errors; (4) better access to information; (5) higher-level support for automation. In addition, another article [64] highlighted the quantitative and qualitative benefits of MBSE: it (1) identifies early design constraints; (2) reduces formal analysis efforts; (3) minimizes errors in manual steps; (4) saves time during the development and production of prototypes. Furthermore, after adopting MBSE in the *Europa Clipper Mission* project, it was observed that "the design is unified, persistent, safely modifiable, and re-usable with better validation of the system model due to wide availability and use" [65]. However, it has been observed that most of the challenges associated with the adoption of MBSE are based on human and technological factors. It starts with awareness and resistance to change at the executive and engineering levels of the organization [66].

The following sections provide a review of automated analysis techniques used in the MBSE environment and a comparison of MBSE tools.

1.3.1 Automated analysis techniques in the MBSE environment

The use of automated analysis techniques allows for reducing manual efforts. Since MBSE enables automated methods, this means that analysis can be accelerated by using one of the available methods.

A comparison of automated analysis techniques in the MBSE environment is provided in Table 1.4 by summarizing sources [67] [68] [69] [70] [71]. The goal of the comparison is to identify and summarize the differences between the available automated techniques. The comparison analyzes five automated techniques by covering their validation rules, metrics, validation-based metrics, requirements verification, and simulation.

Table 1.4. Comparison of automated analysis techniques in MBSE

Description	Usage	Pros	Cons
Validation rules			
Evaluate the architectural model according to the given formula or expression. Validation rules return true and false values. If an error occurs, a message is displayed indicating the cause of the error. Validation rules are divided by criticality into error, warning, and information	▪ Model evaluation (functional, structure, requirements) ▪ Operations / actions control	▪ Flexibility – rules can check different areas of the model, so the application is quite wide-ranging ▪ Clarity – in case of an error, the related element and a message indicating the cause are provided ▪ Reusability – a rule can be applied to other operations, e.g. prohibit model saving if an error occurs	▪ Not user-friendly creation – rule creation, which is especially complex, requires basic programming knowledge to specify the expression or formula of a rule
Metrics			
A quantitative estimation technique is used to assess the status of the current model. Metrics provide numerical information about a specific aspect of a model. The calculation of metrics at regular intervals provide information about the model development	▪ KPI ▪ Status tracking ▪ Model management	▪ Structured results – data is presented in a numerical form ▪ Data analysis – possibility to use various data analysis tools is offered	▪ Not user-friendly creation – metric creation, which is especially complex, requires basic programming knowledge ▪ Lack of clarity – additional data information is required
Validation-based metrics			

Description	Usage	Pros	Cons
A quantitative method that uses a validation expression in a specification. Validation-based metrics combine validation rules and metrics. This method provides numerical information on a specific aspect of the model according to the specified validation rule	▪ Model evaluation ▪ KPI ▪ Status tracking ▪ Model management	▪ Flexibility – as the metrics logic is based on rules, its application to model evaluation covers a wide range of model areas ▪ Clarity – in case of an error, the related element and a message indicating the cause are provided ▪ Structured results – data is represented in the numeric format ▪ Data analysis – possibility to use various data analysis tools is available ▪ User-friendly creation – it is easy to create metrics if validation rules already exist	▪ Time-consuming – if there is no rule, then creating validation-based metrics requires more effort because both rules and metrics need to be created
Requirements verification			
Checks whether the established system requirements are satisfied or not. Requirement verification returns pass, fail, or inconclusive values	▪ Requirement verification	▪ User-friendly – no programming knowledge is required to perform requirement verification ▪ Early requirement verification – early inspection ensures that critical requirements are satisfied	▪ Small elements subset – typically used between requirements and elements of numerical value ▪ Quantitative check – only parameterized requirements are checked
Simulation			

Description	Usage	Pros	Cons
Imitates the real system by simulating various model scenarios and manipulating different values	▪ Model evaluation (functional, structure, requirements)	▪ Analyses support – various analyses are enabled through model simulation ▪ Automation – model simulation is performed with a tool	▪ Difficult in preparing the executable model - it can be quite difficult because the model should strictly comply with specific simulation standards

Based on the comparison results, it can be concluded that the validation rules, validation-based metrics, requirements verification, and simulation are appropriate techniques to automate architecture analysis in the MBSE environment. However, each technique has its own purpose; therefore, it cannot be declared that these techniques are actual alternatives. Validation rules and validation-based metrics are suited to analyze correctness and completeness because they can verify various aspects of the architecture. Requirements verification can be used to verify whether the requirements meet the established needs. The biggest advantage of simulation is that it allows executing a model by running system behavior and performing calculations, which makes the simulation suitable for complex analytical processes. However, preparing an executable model or creating a complex validation rule can be fairly difficult.

Meanwhile, simple metrics are suitable for obtaining structural information about a particular aspect of a model, but, due to the lack of information, this technique is more suitable for tracking the model development process than for performing detailed architectural analysis.

1.3.2 MBSE tool comparison

The goal of the MBSE tool comparison is to identify and summarize their differences. The comparison analyzes four MBSE tools: *Sparx Enterprise Architect (EA) v15.2*, *Catia Magic Systems of Systems Architect (MSoSA) v2021x Refresh2*, *PTC Integrity Modeler (IM) v9.4*, *Unicom System Architect v11.4.9.1 (SyAr)*. The comparison criteria are as follows:

- **ADF support** – a set of supported ADFs. The criterion determines which ADFs are supported by a tool.
- **Automated analysis techniques support** – a set of supported automated analysis techniques. The criterion determines which automated analysis techniques are supported by a tool.
- **Execution semantic** – standard or proprietary. The criterion determines whether the execution semantics is based on simulation standards or is proprietary.
- **Kinds of executable diagrams** – a set of executable diagrams. This criterion determines which of the diagram types can be executed in order to automate the analysis.

The comparison of MBSE tools according to the specified criteria is provided in Table 1.5 by summarizing sources [72], [73], [74], [75], [76].

Table 1.5. MBSE tool comparison

Tool	ADF support	Automated analysis techniques support	Execution semantic	Kinds of executable diagrams
Sparx EA	▪ UAF 1.0 ▪ DoDAF 2.0 ▪ MODAF 1.2 ▪ TOGAF 9.1	▪ Validation rules ▪ Metrics ▪ Simulation	Proprietary	▪ Activity ▪ State machine ▪ Internal block ▪ Sequence ▪ Parametric ▪ BPMN
Catia MSoSA	▪ UAF 1.2 (technical preview) ▪ NAF 4.0 ▪ DoDAF 2.0 ▪ MODAF 1.2 ▪ TOGAF 9.0	▪ Validation rules ▪ Metrics ▪ Validation-based metrics ▪ Requirements verification ▪ Simulation (fUML, PSCS, SCXML, ALF)	Standard	▪ Activity ▪ State machine ▪ Use case ▪ Internal block ▪ Sequence ▪ Parametric ▪ BPMN (limited)
PTC IM	▪ UAF 1.0 ▪ NAF 3.1 ▪ DoDAF 2.0 ▪ MODAF 1.2	▪ Validation rules ▪ Metrics ▪ Simulation	Proprietary	▪ State machine ▪ Internal block ▪ Parametric
Unicom SyAr	▪ UAF 1.0 ▪ NAF 3.0 ▪ DoDAF 2.0 ▪ MODAF 1.2 ▪ TOGAF 9.1	▪ Validation rules ▪ Simulation (BPMN)	Proprietary	▪ BPMN

Based on the comparison results, it can be concluded that the the Catia MSoSA tool supports almost all the latest versions of ADFs, including UAF 1.2, which is scheduled to be released in December 2022; the only version of TOGAF is not the latest. All the compared MBSE tools support simulation and validation rule techniques, but only the Catia MSoSA tool supports validation-based metrics and requirements verification. The Unicom SA tool is the only option that does not support metrics. In addition, only Catia MSoSA uses execution semantic standards; whereas the other MBSE tools have proprietary execution semantics. Finally, most executive diagrams are supported by the Catia MSoSA and Sparx EA tools. Unicom SA only supports the execution of BPMN diagrams, while PTC IM supports the execution of three diagrams: state machine, internal block, and parametric.

1.4 Standard-based execution of UAF architecture models

The foundational subset for executable UML models (fUML) provides a common basis for higher-level UML modeling concepts, as well as an accurate definition of the execution semantics of that subset. The fUML specification covers common behavior, actions, and activities capabilities execution in the behavioral modeling layer [77]. The execution of other UML subsets is defined by two fUML extensions:

- Precise Semantics of UML Composite Structure (PSCS) specification that "defines an extension of fUML syntax and semantics to enable modeling and execution of UML composite structures" [78].
- The Precise Semantics of UML State Machines (PSSM) specification "is an extension of fUML that defines the execution semantics for UML state machines" [79].

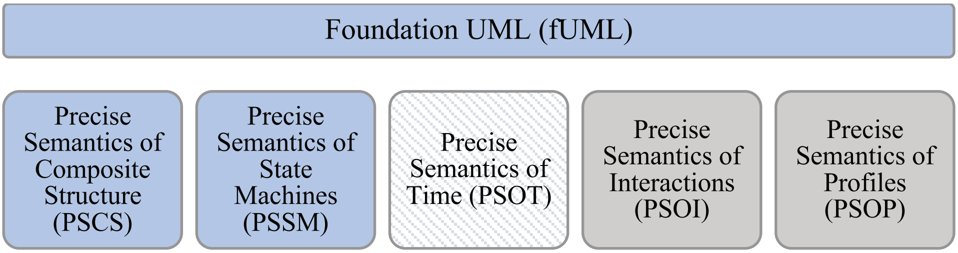

Foundation UML (fUML)
Precise Semantics of Composite Structure (PSCS)
Precise Semantics of State Machines (PSSM)
Precise Semantics of Time (PSOT)
Precise Semantics of Interactions (PSOI)
Precise Semantics of Profiles (PSOP)

Fig. 1.4. Executable UML standards landscape

fUML, PSCS, and PSSM standards together provide a robust foundation for the execution of UML models. In addition, a request for proposal (RFP) is submitted on the *Precise Semantics of Time* (PSOT) for fUML standard in OMG [70]. Fig. 1.4 shows all existing (blue), RFP (stripped gradient fill), and planned (gray) standards related to the execution of UML models.

[18], [80], [81] articles confirm that executing UML models means that the same basis applies to UML extensions such as SysML, UAFP, and others. UAFP is the standard-based implementation that is an extension of UML and SysML profiles. UAF elements, such as *Capability*, *OperationalPerformer*, are inherited from the SysML *Block*. This means that these UAF elements inherit all the features that SysML *Block* has and all the methods of analysis that are based on SysML.

As the Rs domain is a core domain of model execution, the following explains the UAF model execution semantics of activities, states, composite structure, and parametric. The key structural element of the Rs domain is *ResourcePerformer*. *ResourcePerformer* inherits the *Asset* and extends the UML *Class*. The *Asset* inherits the SysML *Block*. This means that *ResourcePerformer* inherits all UML *Class* and SysML *Block* features combined.

Activities

The *ResourcePerformer* is capable of performing *Functions*. *Function* extends the UML *Activity*, and *IsCapableToPerform* relationship extends the UML *Abstraction* relationship and inherits the SysML *Allocation* relationship. To set a

Function as a *ResourcePerformer* classifier behavior, it must be directly owned. The *Function's* semantics are the same as those of UML *Activities*.

FunctionAction is an element that calls a *Function* in a different *Function* context. It extends UML *CallBehaviorAction*. *Actions* are connected with *FunctionEdges*: *FunctionControlFlow* and *FunctionObjectFlow*, where *FunctionControlFlow* extends the UML *ControlFlow*, and *FunctionObjectFlow* extends the UML *ObjectFlow*. The difference in UAF is that *FunctionEdge* can realize *ResourceExchanges* that show how *Functions* produce and consume resources exchanged by their performers. Although fUML and its extensions do not support *ResourceExchanges*, pins can be created and connected to objects by using *ObjectFlows*. Besides the limitation mentioned above, the UAF process model is compliant with the semantics of UML *Activities* and can be executed with fUML.

States

In the UAF, the *Resources States* viewpoint defines only one element called *ResourceStateDescription* that extends the UML *State Machine*. *ResourceStateDescription* is owned by *ResourcePerformer* and can specify the behavior of the classifier. There are no restrictions or extensions that would affect the semantics of the execution of the state machines defined in PSSM.

The *States* in the UAF can be combined with *Functions* that can define actions and transition effects. In this case, *Function* does not need to belong directly to the *ResourcePerformers*.

Composite Structures

The UAF extensions define UML *Properties*, *Ports*, and *Connectors*. *ResourceRole* extends the UML *Property*, *ResourcePort* extends the UML *Port*, inherits SysML *ProxyPort*, and *ResourceConnector* extends the UML *Connector*.

Since *ResourcePort* inherits SysML *ProxyPort*, its semantics are different from those described in PSCS [78]. As part of the SysML 1.7 specification, it is planned to define the extension of PSCS to support the semantics of certain SysML concepts, such as *ProxyPorts*. Currently, UML semantics can be used to execute UAF composite structure models, but there are some limitations.

Parametric

UAFP was specially designed to support SysML *Parametric*. Therefore, the only requirement of UAFP was to perform SysML parametric analysis within the context of elements that have been inherited from the SysML *Block*. For example, *ResourcePerformer* belongs to this category and can be used for parametric analysis in the *Resource* domain. In SysML, parametric analysis is based on *Block*-owned *Value*. *Values* are linked to *ConstraintBlock Parameters* to evaluate constraints expressions of *ConstraintBlock*.

Table 1.6 provides a list of the execution limitations of UAF models using existing standard-based techniques, along with the possible workarounds. The provided workarounds are validated in article [18].

Table 1.6. Limitations of UAF models execution [18]

Limitations	Workarounds
Activities are not defined as performed activities in accordance with the IsCapableTo Perform relationship	Activity must belong directly to the classifier
Exchanges are not supported in Activity diagrams	It is necessary to model the input and output pins
The semantics of SysML *ProxyPort* are not supported by existing simulation standards	*ProxyPorts* are treated as UML ports
Exchanges are not supported in the type of the composite structure diagram	It is necessary to use UML ports
There is no support for continuous flow simulation	There is no easy workaround. Manual loops in the activity diagram and value properties for capacity management should be introduced

Based on the Table 1.6 results, it can be concluded that UAF models can be executed with the existing standard-based approaches with certain limitations requiring workarounds. Furthermore, the combination of simulation standards and modeling standards provides a powerful basis for analyzing and simulating SoS models.

1.5 Trade study processes and methods

The trade study is a common task in the daily routine of SE. Trade studies are carried out during the phases of concept definition, development, and design so that to select the operational concepts, the initial capabilities, and the requirements for the high-level system architecture.

There are several definitions of the trade study in the literature:

- *Definition 1*: "Trade study as analysis that focuses on ways to improve systems performance on some highly important objective while maintaining system's capability in other objectives." [82]
- *Definition 2*: "Trade Studies is the System Engineering element that multidisciplinary teams use to identify the most balanced technical solutions among a set of proposed viable solutions. It is a key tool in developing designs that meet stakeholder requirements in the most cost-efficient manner possible." [83]
- *Definition 3*: "Trade studies are decision-making activities used to identify the most acceptable technical solution among a set of proposed solutions." [1]
- *Definition 4*: "Trade study is a formal decision-making methodology used by integrated teams to make choices and resolve conflicts during the systems engineering process." [84]
- *Definition 5*: "*A* trade study is a decision-making method used to identify the best solution among a group of proposed solutions." [28]

The trade study is a formal tool to support decision-making. A comparison of alternative solutions determines whether or not they meet certain criteria of the preferred solution. A trade study is an objective comparison of the performance, costs, timetable, risks, and other reasonable criteria of the alternative.

Trade studies aim at supporting decision-making throughout the system's lifecycle and providing an objective basis for the selection of one or more alternative solutions of an engineering problem. Once a set of alternatives has been identified, a trade study team analyzes the alternatives by using a series of decision-making criteria. These criteria are 'traded' to determine the most balanced and recommended alternative. Additionally, a trade study records the requirements, assumptions, criteria, and priorities used in making decisions. This is useful, as new information is often encountered, and decisions are therefore re-evaluated [1].

The following sections present the trade study processes and evaluation methods.

1.5.1 Trade study process

The worldwide SE community has acknowledged and is currently using various trade study processes. This section describes and analyzes the most popular ones that are introduced by NASA, NAF, Reiter, and MITRE.

Trade study process by NASA

NASA introduced the trade study process in 2016 which assists the decision-maker in selecting the most suitable alternative from a set of several viable alternatives [85]. The whole trade study process involves twelve steps (Fig. 1.5).

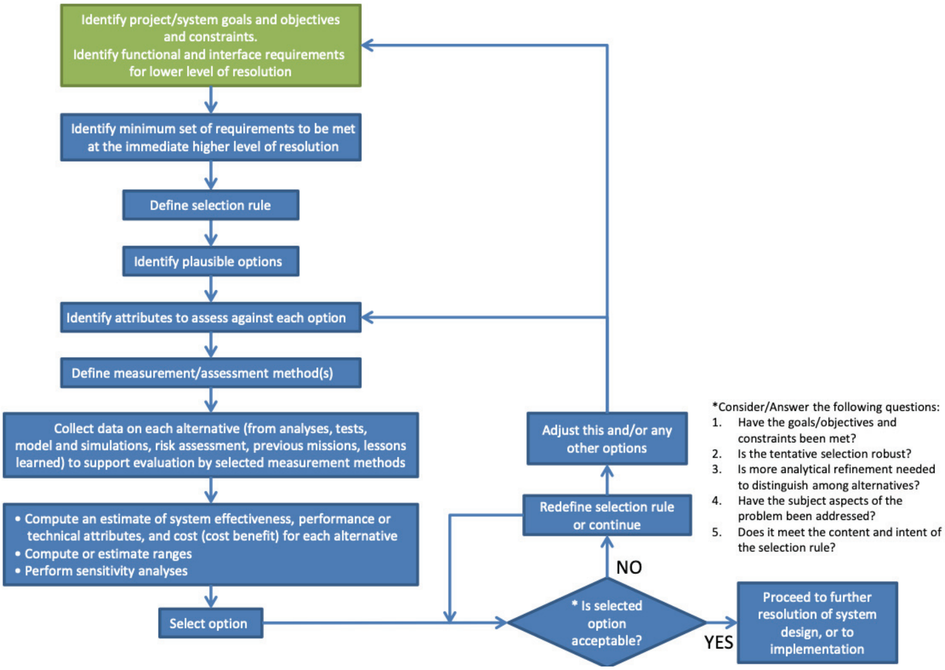

Identify project/system goals and objectives and constraints. Identify functional and interface requirements for lower level of resolution
Identify minimum set of requirements to be met at the immediate higher level of resolution
Define selection rule
Identify plausible options
Identify attributes to assess against each option
Define measurement/assessment method(s)
Collect data on each alternative (from analyses, tests, model and simulations, risk assessment, previous missions, lessons learned) to support evaluation by selected measurement methods
• Compute an estimate of system effectiveness, performance or technical attributes, and cost (cost benefit) for each alternative
• Compute or estimate ranges
• Perform sensitivity analyses
Select option
* Is selected option acceptable?
NO
YES
Redefine selection rule or continue
Adjust this and/or any other options
Proceed to further resolution of system design, or to implementation
*Consider/Answer the following questions:
1. Have the goals/objectives and constraints been met?
2. Is the tentative selection robust?
3. Is more analytical refinement needed to distinguish among alternatives?
4. Have the subject aspects of the problem been addressed?
5. Does it meet the content and intent of the selection rule?

Fig. 1.5. Trade study process introduced by NASA [85]

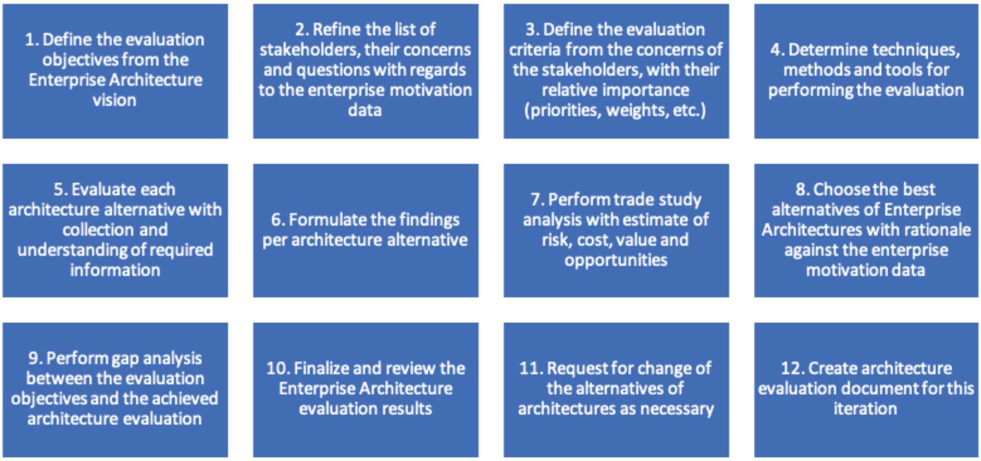
The proposed trade study process begins by acquiring knowledge and understanding the project or system goals, objectives, and constraints as they may be affected by the results of the trade studies. Functional analyses are conducted to understand the full impact of the objectives and to formulate the appropriate system analyses. Then, a set of plausible alternatives is defined that may achieve the goals and objectives of the system. Furthermore, a set of selection rules is defined which are expected to clearly determine how the variables of the results will be used to select the preferred alternative.

Next, the analytical portion of trade studies begins, which consists of three steps. First, the system's effectiveness, performance, and cost measures along with the applicable measurement methods are defined. Second, the data on each alternative is collected to support the evaluation of the measurements by the chosen measurement methods. Third, each alternative is evaluated and calculated on the basis of the indicated measures, while also including the uncertainty range and sensitivity analysis.

After the evaluation of the alternatives, the person or group responsible should make a tentative selection. If the tentative selection is not acceptable, the trade study must be repeated.

Trade study process by NAF

NAF specification describes a wide range of activities of the architect in creation, implementation, and management of the architecture [40]. The trade study is one of the presented processes which consists of twelve steps (Fig. 1.6).



1. Define the evaluation objectives from the Enterprise Architecture vision

2. Refine the list of stakeholders, their concerns and questions with regards to the enterprise motivation data

3. Define the evaluation criteria from the concerns of the stakeholders, with their relative importance (priorities, weights, etc.)

4. Determine techniques, methods and tools for performing the evaluation

5. Evaluate each architecture alternative with collection and understanding of required information

6. Formulate the findings per architecture alternative

7. Perform trade study analysis with estimate of risk, cost, value and opportunities

8. Choose the best alternatives of Enterprise Architectures with rationale against the enterprise motivation data

9. Perform gap analysis between the evaluation objectives and the achieved architecture evaluation

10. Finalize and review the Enterprise Architecture evaluation results

11. Request for change of the alternatives of architectures as necessary

12. Create architecture evaluation document for this iteration

Fig. 1.6. Trade study process by NAF

The proposed trade study process begins with the definition of the evaluation objectives and the refinement of the list of the stakeholders. Then, according to the interests of the stakeholders, the evaluation criteria are defined by priority and weight. Accordingly, the technologies, methods, and tools used for the evaluation are determined. After the previous step has been completed, each architecture alternative is assessed according to the specified criteria. Then, a trade study analysis is conducted to assess the risks, costs, and opportunities.

Finally, the responsible person chooses the best alternative and presents the justification. Then, the three final activities are carried out: (1) the gap analysis between the objectives of the evaluation and the results obtained from the evaluation; (2) the finalization and review of the results of the evaluation; (3) documentation of the architecture evaluation.

Trade study process by J. Reiter

Reiter introduced a trade study process in 2016 to compare several alternatives concerning the performance, cost, schedule, risk, and all other reasonable criteria of all realistic alternative requirements [86]. The whole trade study process evolves in seven main steps (Fig. 1.7).

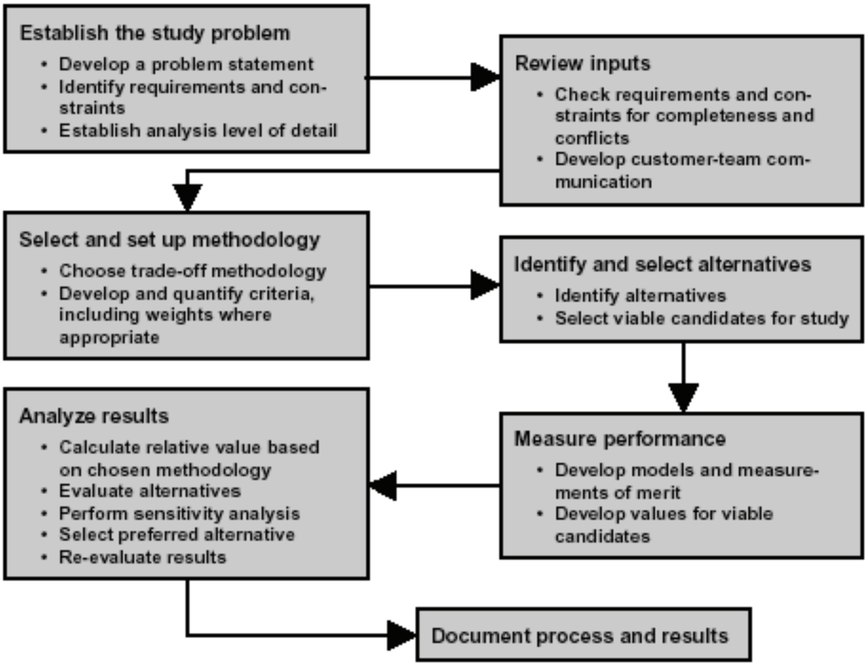

Establish the study problem
• Develop a problem statement
• Identify requirements and constraints
• Establish analysis level of detail
Review inputs
• Check requirements and constraints for completeness and conflicts
• Develop customer-team communication
Select and set up methodology
• Choose trade-off methodology
• Develop and quantify criteria, including weights where appropriate
Identify and select alternatives
• Identify alternatives
• Select viable candidates for study
Analyze results
• Calculate relative value based on chosen methodology
• Evaluate alternatives
• Perform sensitivity analysis
• Select preferred alternative
• Re-evaluate results
Measure performance
• Develop models and measurements of merit
• Develop values for viable candidates
Document process and results

Fig. 1.7. Trade study process introduced by Reiter [86]

First, the problem is established by determining the necessary information, requirements, constraints, and the detailed level of analysis to be needed. Then, requirements and constraints are checked to ensure the accuracy and conflict while communication with the customers is being established. Third, the trade study method, criteria, and the corresponding weighting systems are determined. Afterward, all the alternatives are identified, and potential candidates are selected for the study. The model is then developed to measure the quality of each design and assign a value to each alternative. Then, the result is analyzed, based on the chosen method, calculated values, evaluated alternatives, and performed sensitivity analysis. The results of the trade study are recorded.

Trade study process by MITRE

The MITRE organization presented a trade study process in 2021, which leads to a justifiable and objective trade study [87]. The whole trade study process involves nine principal steps (Fig. 1.8).

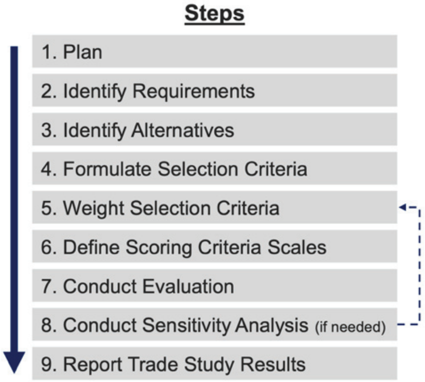

Steps
1. Plan
2. Identify Requirements
3. Identify Alternatives
4. Formulate Selection Criteria
5. Weight Selection Criteria
6. Define Scoring Criteria Scales
7. Conduct Evaluation
8. Conduct Sensitivity Analysis (if needed)
9. Report Trade Study Results

Fig. 1.8. Trade study process introduced by MITRE [87]

First, the scope of the study is determined by identifying the available resources, time, environments, and other parameters required prior to the study [87]. After determining the scope of the trade study, the trade study plan must be documented and agreed with the supporting organization. Then, the requirements to be used as a basis for the study are identified. It shall also include any design constraints that bound the selection of alternatives and the criteria used to conduct the evaluation. Third, possible solution alternatives are identified. Alternatives should be sufficiently similar to enable a relevant comparison using criteria and the associated measures. Fourth, a set of the trade study criteria is developed. The criteria should be applied to all the alternatives and be relevant to the requirements. Next, the relative weights and scoring scales are assigned to each criterion.

Once all the necessary data has been identified, the evaluation of the alternatives follows. Evaluation data is converted into total scores for each alternative by using scales and weights. If necessary, sensitivity analysis may be carried out to examine the scores and criteria for each alternative. Finally, all the data must be documented with results and recommendations. Recommendations should be based on the written basis and include one alternative or other permutations.

A review of the trade study processes showed that most of the existing trade study processes: (1) do not provide the formally defined trade study process; (2) define only the main flow of the trade study excluding details of alternatives evaluation, such as specific evaluation methods, techniques with algorithms or constraints; (3) are not adapted to the domain; therefore, they provide only general steps and use general terminology; (4) do not provide any guidance how to apply the trade study process in the MBSE environment; (5) do not define roles or input/output data for each process step.

The following sections present the evaluation methods used in the trade study process.

1.5.2 Multi-criteria decision analysis methods

Multi-criteria decision analysis (MCDA) is an instrument that supports decision makers facing numerous and conflicting assessments [88]. The aim of MCDA is to highlight conflicts between alternatives and find a way to reach compromises in a transparent process. Additionally, this approach is highly flexible, and the results can be quantified in non-monetary terms and expressed in ordinary or numerical terms [89].

MCDA is a relevant research object among the scientific community. Thesis by Morkevičius [47] proposed a method for aligning business and information systems based on enterprise architecture models modeled on the UPDM principles. Business and information systems are aligned by multiple criteria to identify incompleteness, thus helping to achieve a higher level of the quality of business and information systems alignment. The paper [90] described the system for security management based on the MCDA principles. The paper [91] developed a method that uses a mixed selection strategy for MCDA under complex uncertainty.

Currently, several methods can be used to implement MCDA, but each method varies in how the stakeholders' inputs are collected and handled. This section briefly describes five most common MCDA methods: the *Multi Attribute Utility Theory* (MAUT), the *Pugh method*, *Multivariate analysis* (MVA), the *Analytic Hierarchy Process* (AHP), and the *Technique for Order Preferences by Similarity to Ideal Solutions* (TOPSIS).

MAUT

MAUT is a technique for supporting decision-making when decision-makers have to choose from a limited range of available alternatives [92]. In MAUT, the overall assessment of the alternatives is defined as weighted addition of their values relative to their relevant attributes. This technique requires decision makers to evaluate the alternatives separately for each attribute. For example, decision-makers assign the relative weights to various attributes. Then, the value and the weight are combined and grouped by the formal model, which generates a comprehensive assessment of each alternative [93].

The linear additive preference function is mainly used, but other functions can also be used, for example, the Keeney and Raiffa function [94]. Keeney and Raiffa and Von Winterfeldt and Edwards are important researchers in this field.

Although the actual application of MAUT varies, all of these procedures include the following five steps [93]:

1. Define alternatives and value-relevant attributes;
2. Evaluate each alternative separately on each attribute;
3. Assign relative weights to the attributes;
4. Aggregate the weights of the attributes and single attribute evaluations of the alternatives to obtain an overall evaluation of alternatives;
5. Perform sensitivity analyses and make recommendations.

Pugh method or decision-matrix method

Stuart Pugh introduced the Pugh method in 1981 [95]. This method is a qualitative selection technique using a range of alternatives to choose the best solution [96]. It helps to determine which solution is more valuable than the other options.

The Pugh matrix is a priority matrix comparing the alternative solutions to the already existing solutions or to the benchmarks that will be achieved in the near future [97]. Engineers are using the Pugh matrix extensively to improve and facilitate the selection process of structured concepts [98]. Wurthmann observed that the Pugh matrix can maximize the potential of innovative solutions in decision-making and creativity [96].

MVA

MVA is a series of techniques that are used to analyze a dataset containing more than one variable [99] [100]. Therefore, it is strongly recommended to use MVA to investigate the interrelationship between different attributes of the data so that to identify groups and correlate the underlying data with the graphics based on this analysis [101] [102] [103]. Initially, the multivariate analysis technology was used in statistics to reveal causal relationships. Today, the MVA methods are used in a variety of fields: economy, data mining, linguistics, etc.

The MVA technique is divided into two groups according to the relationship with the data: dependence and interdependence. Dependence methods are used when one or more variables depend on another variable by looking at the cause and the effect. The dependency method includes such techniques as multiple regression, conjoint analysis, multiple discriminant analysis, linear probability models, multivariate analysis of variance, structural equation modeling, and canonical correlation analysis.

The interdependence method consists of understanding the structure and the underlying patterns of the dataset. In this case, there is no variable dependent on other variables, and thus the causal relationship is not considered. In contrast, interdependence methods are aimed at giving meaning to a set of variables or grouping them in a meaningful way. The interdependence methods include factor analysis, cluster analysis, and correspondence analysis.

The main advantage of MVA is that it takes into account several factors. It examines different independent variables influencing the dependent variables. However, MVA requires more complex computations to obtain the answers. The governance and preparation required for MVA are usually much more complex and time-consuming.

AHP

AHP was introduced in 1970 by Thomas L. Saty. AHP is a multi-attribute methodology providing proven and effective ways to address complex decision making [104]. This method helps to identify and weigh the selection criteria, analyzes the data collected for these criteria, and accelerates decision-making processes [88]. AHP shows the decision of compatibility and incompatibility: the remuneration of multi-criteria decision-making [105] [106]. Furthermore, paired comparison facilitates judgment and calculation. AHP provides the appropriate decision-making framework by quantifying its criteria and alternatives and by linking them to the overall objective [107]. Typically, the AHP process contains five main steps:

1. Define the goal of the decision;
2. Structure the decision problem in a hierarchy;
3. Pair comparison of criteria in each category;
4. Calculate the priorities and the consistency index;
5. Evaluate the alternatives according to the priorities identified.

AHP is widespread in the scientific community and is associated with such distinct fields as engineering, medicine, and other sciences. The main advantage of AHP is its usability. AHP solves tedious problems by breaking them down into smaller steps [108]. Furthermore, authentic information sets do not need to be supplied. The structure of AHP provides an easy way for a researcher to deal with complex issues. However, AHP uses accurate properties for its solutions and can identify only direct models whose output specifically matches its information [109].

TOPSIS

TOPSIS was introduced in 1981 by Ching-Lau Hwang and Yoo as the MCDA method [110]. "TOPSIS is based on the fundamental premise that the best solution has the shortest distance from the positive-ideal solution, and the longest distance from the negative-ideal one. Alternatives are ranked with the use of an overall index calculated based on the distances from the ideal solutions" [111].

The main strengths of the TOPSIS method are simplicity, rationality, good computational efficiency, and the ability to measure the relative performance of each alternative in a simple mathematical form [112]. The main weaknesses of TOPSIS are that the use of the *Euclidean Distance* does not take into account the correlation of attributes and does not provide a way of ensuring consistency in the weight control [113].

A comparison of the five MCDA methods is presented in Table 1.7 by covering their main strengths and weaknesses.

Table 1.7. Summary of MCDM methods

Method	Strengths	Weaknesses
MAUT	▪ Takes uncertainty into account ▪ Can incorporate preferences	▪ Needs a lot of input data ▪ Preferences need to be precise
Pugh	▪ Simplicity ▪ Does not require a large set of data	▪ Value assignments are subjective ▪ Criteria list is arbitrary
MVA	▪ Examines different independent variables influencing dependent variables ▪ Easy to perform with statistical packages	▪ Requires complex computations and high-level mathematics ▪ Requires a large set of data
AHP	▪ Simplicity ▪ Scalability	▪ Interdependence of criteria and alternatives can result in

Method	Strengths	Weaknesses
	▪ The hierarchy structure can be easily adjusted to adapt to many differently-sized problems ▪ Not data intensive	inconsistencies between judgment and ranking criteria ▪ Experts' usage to determine weights
TOPSIS	▪ Simplicity ▪ Computational efficiency ▪ Ability to use mathematical forms to measure alternatives	▪ The use of Euclidean distance does not consider the correlation of attributes ▪ Difficult to weigh and maintain the coherence of judgment

Analysis of the MCDA methods disclosed that: (1) Pugh, AHP, and TOPSIS methods are easy to apply; (2) MVA is easy to apply, but only when statistical packages are used; (3) MAUT and MVA methods require a large set of input data and complex computations including high-level mathematics; (4) Pugh and AHP are not data intensive; (5) AHP is the only method that could be used with a large set of criteria due to its scalability and structure.

1.5.3 Sensitivity analysis

Sensitivity analysis (SA) is a method studying the effects of input uncertainty on the response to the model output [114] [115] [116]. Sensitivity analysis is used to determine the reliability of a system result and to understand the relationship between the variables and their relative influence on the performance of the system [117] [118]. If there are significant output differences when changing a range of the input variables, the output is sensitive. If the output does not change significantly, then, the output is insensitive or robust. Sensitivity analysis is a common method used in a variety of fields [119] [120] [121], including economy, multi-criteria decision making, engineering, chemistry, and others.

The methods of sensitivity analysis are divided into two main groups: local sensitivity analysis and global sensitivity analysis.

Local Sensitivity Analysis

Local SA is an approach for assessing changes in the model output based on the changes in the input of a single parameter [122] [123]. The input parameter is changed one at a time, and the other parameters are kept unchanged. Typically, a local SA is used if the output of a model is linearly linked to parameters near a particular nominal value [122]. Local sensitivity is a great tool when calibrating models. It can help determine which parameters should be changed to ensure that the system reproduces the desired result.

The main limitation of local sensitivity analysis is that it does not allow evaluation of all model parameters simultaneously, since it only examines small parts of the design [124]. Furthermore, interactions between parameters cannot be evaluated. However, local SA methods are preferable to large systems because they require less computing power than the global SA.

Local SA can be carried out by using several possible methods, such as:

- The *One-at-a-time (OAT)* method involves setting and changing one parameter at a time while all other parameters remain fixed [125]. It changes an input parameter (X_i) and measures the impact it has on the output (Y). Typically, parameters are changed by 20% [126] [127]. The sensitivity by the OAT method is calculated on the basis of Equation (1).

$$sensitivity = \frac{\Delta Y}{\Delta X_i} \tag{1}$$

- The *Partial derivatives* method involves a partial derivative of the output factor (Y) in relation to the input factor X_i [128]. The sensitivity by the partial derivative method is calculated on the basis of Equation (2).

$$sensitivity = \frac{\partial Y}{\partial X_i} \tag{2}$$

- The *Sensitivity index* is a number calculated by defined algorithms that provide information about the relative sensitivity of the results of different parameters in the model [129]. The higher is the sensitivity index, the higher is the sensitivity of the result to the changes in that parameter. The sensitivity index is calculated on the basis of Equation (3).

$$sensitivity = \frac{Y_{max} - Y_{min}}{Y_{max}} \tag{3}$$

Global Sensitivity Analysis

Global SA is an approach to simultaneously changing all the parameters in the parameter space [122]. This allows simultaneous evaluation of the relative contribution of each parameter and the interaction between parameters with the deviation in the model output. Global SA focuses on variations in the model output and determines how the input parameters affect the output parameters. It provides a quantitative and rigorous view of how different inputs affect the output.

Global SA can be used without knowing the unknown parameters and performed before the calibration process of the model [130]. Additionally, global SA can be used to reduce parameters when the results of particular parameters have no impact on the change. In general, global SA is preferred, if possible, for being more detailed. However, large-scale systems require higher computational power, thus making SA prohibitively expensive.

There are various global SA methods available, such as the Sobol method, Fourier amplitude sensitivity analysis (FAST), multiparametric sensitivity analysis (MPSA), and partial rank correlation coefficient (PRCC). [131] paper provides the comparison of these global SA methods by including their essential features, advantages, and limitations.

Analysis of local and global SA methods, about which, we can disclose that: (1) the local SA determines sensitivity to changes in a single parameter value, while the global SA examines sensitivity to the whole distribution of parameters; (2) the global SA for large systems is very expensive in terms of the computational power, while the local SA requires less computational power.

1.6 Analysis conclusions

Conclusions for this chapter are as follows:

1. Analysis of ADFs revealed that the currently existing ADFs do not define how a trade study for SoS architectures should be conducted, nor do they provide a formalized trade study process. ADFs provide only a basis for collecting the data required for trade studies from the architecture point of view. However, no concepts are provided that are needed to specify the specific trade study data, such as a set of the selection criteria or the priority level of a specific criterion.
2. Analysis of the automated analysis techniques, including the execution of standard-based UAF models, revealed that: (1) validation rules, validation-based metrics, requirements verification, and simulation are appropriate techniques to automate architecture analysis in the MBSE environment. However, each technique has its own purpose and the specific scope in architecture analysis; (2) fUML, PSCS, and PSSM standards together provide a robust foundation for the execution of UML models. As UAFP is an extension of the UML and SysML profiles, it can be concluded that UAF models can be executed with the already existing standard-based approaches. However, there are some limitations that require the application of certain workarounds.
3. Analysis of MBSE, UAF, and fUML revealed that they are suitable means for automating evaluations of alternative SoS architectures in the trade study process. However, extensions of the UAFP metamodel are necessary to perform a trade study in the MBSE environment, as it lacks specific concepts to define and perform a trade study in MBSE, along with UAF and fUML.
4. Analysis of the trade study processes showed that the majority of the existing trade study processes: (1) do not provide a formally defined trade study process; (2) define only the main flow of a trade study, while excluding details of alternative evaluation, such as specific evaluation methods, techniques with algorithms or constraints; (3) are not adapted to the domain; therefore, they provide only general steps and use general terminology; (4) do not provide any guidance on how to apply the trade study process in the MBSE environment; (5) do not define the roles or input/output data for each process step.
5. Analysis of the MCDA and SA methods disclosed that: (1) most MCDA methods are easy to apply; however, their use is hampered by the need to provide a large set of input data and the application of complex computations; (2) AHP is the only method that is easy to apply, not data intensive, and can be used in a large set of criteria due to its scalability and structure; (3) the local SA is appropriate when the sensitivity of a particular selection criterion is sought. Meanwhile, performing a global SA would change several selection criteria at the same time, and therefore it would not be possible to obtain accurate data on the sensitivity of a specific selection criterion.

2 UT3SA: UAF-BASED TRADE STUDY METHOD FOR SoS ARCHITECTURES

The UT3SA method aims to guide SoS engineers through the trade study process in the MBSE environment. This method helps to indicate the most preferred alternative solution among several alternatives by performing various analyses. In order to autonomize the analysis in the MBSE environment, the validation-based metrics, requirements verification and simulation techniques are chosen. Furthermore, the UT3SA method provides guidelines on how to prepare the executable model in the MBSE environment for an automated trade study.

The UT3SA method is based on the SoS architecture modeled by using the UAF principles. The method is constituted of two main parts (Fig. 2.1): (1) the UAF-based trade study process; (2) analysis algorithms and guidelines in the MBSE environment.

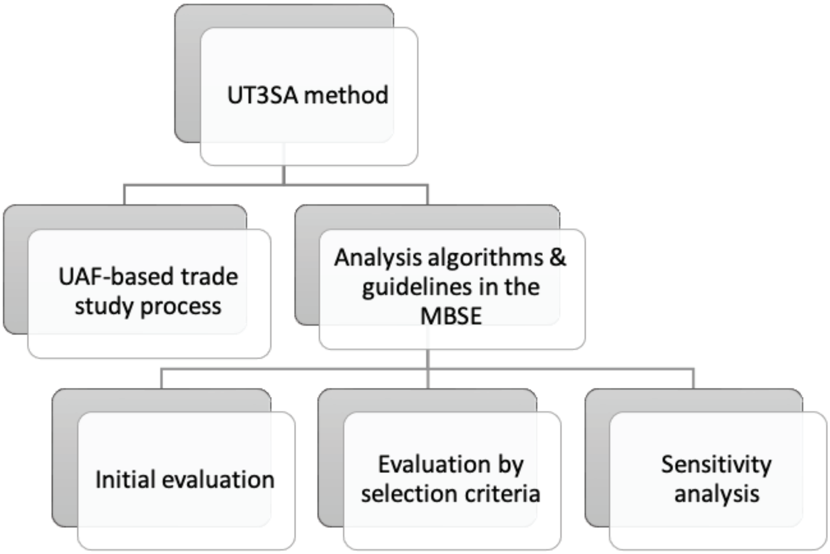

UT3SA method
UAF-based trade study process
Analysis algorithms & guidelines in the MBSE
Initial evaluation
Evaluation by selection criteria
Sensitivity analysis

Fig. 2.1. Structure of the UT3SA method

The UT3SA method summarizes the currently available trade study processes and links them with the UAF that is applicable to many domains rather than their accosters: MODAF, DoDAF, or NAF. In order to simplify analyses of the trade study process, UT3SA provides guidelines and algorithms to perform three specific analyses: (1) initial evaluation; (2) evaluation algorithm by selection criteria; (3) sensitivity analysis.

This chapter includes five parts. The first part describes the UAF-based trade study process. The second part introduces techniques for the initial evaluation in order to define the quality of the architecture of each alternative solution. The third part presents the method for evaluating alternative solution architectures and performing sensitivity analysis in the MBSE environment. The fourth part provides a comparison of the UT3SA method to other trade study processes and methods. The fifth part summarizes the UT3SA method.

2.1 UAF-based trade study process

The presently introduced trade study process has been designed to incorporate automated methods while performing a trade study. The process is fully compatible with UAFP. Furthermore, the proposed trade study process includes the necessary steps when a trade study is conducted as a result of an acquisition process.

The trade study process involves five main roles:

- **SoS Engineer** defines the preferred SoS logical architecture, as well as determines the capabilities of and the requirements for SoS. The SoS engineer models the views of two UAF domains: *Operational* (Op) and *Strategic* (St).
- **Trade Study Lead** manages the entire trade study process. The trade study lead initiates the process, brings together experts from multidisciplinary fields in a trade study team, and identifies the crucial capabilities and operational requirements of the preferred SoS that must be met. Additionally, this role is responsible for preparing reports and communicating with contractors.
- **Trade Study Experts Group** establishes the selection criteria and weights that are used to evaluate the architecture of each alternative SoS solution. In addition, this role gathers the necessary data for each alternative SoS architecture and summarizes the results of the two key evaluations: *Base Check* and *Deep Check*. The trade study experts group incorporates two sub-roles: the expert and the chief expert. The expert role is responsible for establishing the selection criteria and assigning the weights, meanwhile, the chief expert role is responsible for reviewing and approving the set of the selection criteria and the Base Check final results.
- **Decision-Making Authority** makes the final decision after the two main evaluations: the Base Check and the Deep Check. This role may also review the defined key capabilities, operational requirements, and measurements, as well as review the terms due to the acquisition process.
- **Contractor** prepares the solution architecture based on the preferred SoS requirements. The contractor models the Resource domain (Rs) views of UAF.

Fig. 2.2 presents the proposed trade study process which consists of ten main stages. At each stage of the process, the role that performs the process is assigned, the input/output data is defined, the occurrence is identified, and the process order is set. The following figure is a detailed explanation of all stages of the UAF-based trade study process.

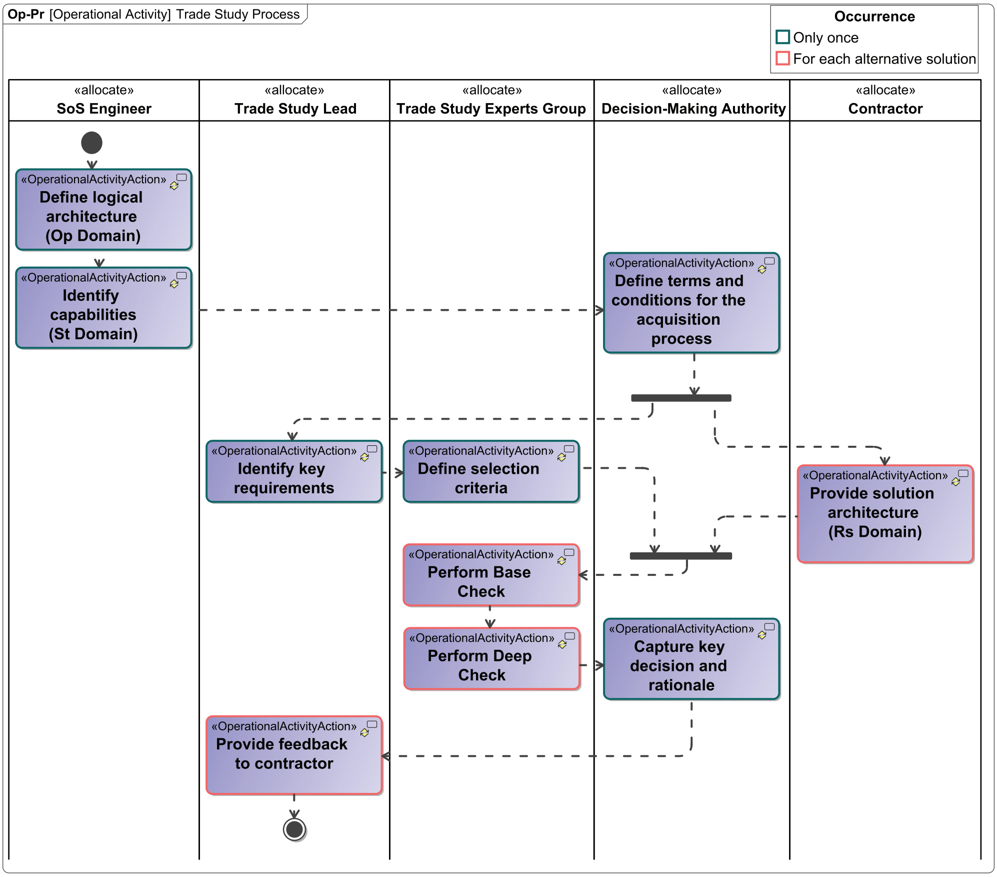

Op-Pr [Operational Activity] Trade Study Process
Occurrence
□ Only once
□ For each alternative solution
«allocate» SoS Engineer
«allocate» Trade Study Lead
«allocate» Trade Study Experts Group
«allocate» Decision-Making Authority
«allocate» Contractor
«OperationalActivityAction» Define logical architecture (Op Domain)
«OperationalActivityAction» Identify capabilities (St Domain)
«OperationalActivityAction» Define terms and conditions for the acquisition process
«OperationalActivityAction» Identify key requirements
«OperationalActivityAction» Define selection criteria
«OperationalActivityAction» Provide solution architecture (Rs Domain)
«OperationalActivityAction» Perform Base Check
«OperationalActivityAction» Perform Deep Check
«OperationalActivityAction» Capture key decision and rationale
«OperationalActivityAction» Provide feedback to contractor

Fig. 2.2. UAF-based trade study process

Define logical architecture (Op domain)

Logical architecture is one of the three main abstraction levels of ADF defining detailed functional requirements. This stage indicates the 'what' that must be achieved by the preferred solution [132]. The Op domain defines the tasks and activities necessary to perform operations and meet the needs and expectations of the stakeholders.

In this stage, the SoS engineer models the UAF Op domain views to describe the preferred SoS logical architecture. In addition, the SoS engineer also sets out what SoS aims to accomplish and who is responsible for it.

Responsible role: SoS Engineer
In: Stakeholder needs
Out: Logical architecture (Op domain view model)
Occurrence: Only once

Identify capabilities (St domain)

In order to achieve a strategic vision, capabilities with certain functions and implementation for resources are identified. "Capability is the ability of an enterprise to achieve a desired system along with specified measures" [17].

In this stage, the SoS engineer models the St domain views of UAF. The set of identified capabilities and capability requirements specifies what characteristics they will feature and at what level so that to meet the identified requirements of the stakeholders. In addition, traceability links are established between the capabilities and the operational requirements.

Responsible role: SoS Engineer
In: Op domain view model
Out: St domain view model with traceability links to Op domain model
Occurrence: Only once

Define terms and conditions for the acquisition process

When a trade study is carried out as a result of an acquisition process, the terms and conditions should be defined in accordance with the applicable government regulations. It is essential to ensure that all contractors have a level playing field.

In this stage, terms and conditions must be defined, such as the requirements for the contributors, the requirements for the solution, the key dates, and other points. Regardless of whether an acquisition process is involved in a trade study, it is recommended to define the internal rules for communication, the responsible people, the selection procedures, and other criteria.

Responsible role: Decision-Making authority
In: Government regulations
Out: Terms and conditions for the acquisition process
Occurrence: Only once

Identify key requirements

The preferred solution includes a number of capabilities, operational requirements and measurements that must be satisfied in order to achieve the expected functionality. However, due to time and resource constraints, it is important to ensure that the chosen solution possesses the critical features.

In this stage, the trade study lead identifies the key capabilities, the operational requirements and the quantitative measurements that must be satisfied. In addition, it can be prioritized. Prioritization is useful if no alternative can meet all the crucial requirements.

Responsible role: Trade study lead
In: St and Op domain models
Out: Key capabilities (St domain), operational requirements (Op domain), quantitative measurements
Occurrence: Only once

Provide solution architecture (Rs domain)

Once the acquisition process begins, contractors can get familiarized with the terms and requirements, and decide whether they will participate in the acquisition. A confidentiality disclosure agreement can be signed if necessary.

In this stage, each contractor models the Rs domain views of UAF. The contractors have access to the St and Op domain models to prepare their solution architecture based on the preferred SoS requirements. Each contractor should submit its solution architecture prior to the specified deadline.

Additionally, if a trade study is not involved in an acquisition process, a set of alternative solution architectures should be identified at this stage.

Responsible role: Contractor
In: Terms and conditions for the acquisition process; Op and St domain models
Out: Set of Rs domain models with traceability links to St and Op domain models
Occurrence: For each alternative solution architecture

Define Selection Criteria

The selection criteria must be clearly set out how the outcome variables will be used to choose the preferred alternative solution architecture. A set of selection criteria is determined by taking into account the domain of the preferred system, its critical aspects, and the stakeholders' needs.

In this step, the group of trade study experts defines a set of the selection criteria (Fig. 2.3) to be used for the evaluation of each plausible SoS solution architectures. Each selection criterion should have a weight assigned that would reflect its criticality.

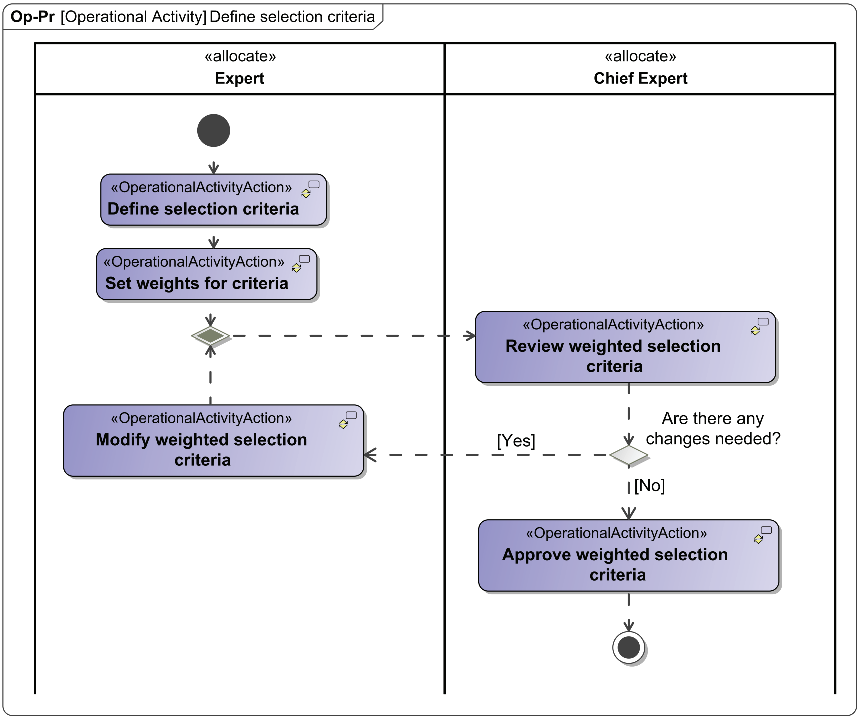

Op-Pr [Operational Activity] Define selection criteria
«allocate»
Expert
«allocate»
Chief Expert
«OperationalActivityAction»
Define selection criteria
«OperationalActivityAction»
Set weights for criteria
«OperationalActivityAction»
Review weighted selection criteria
Are there any changes needed?
[Yes]
[No]
«OperationalActivityAction»
Modify weighted selection criteria
«OperationalActivityAction»
Approve weighted selection criteria

Fig. 2.3. UAF-based trade study subprocess – Define selection criteria

Responsible role: Trade study expert and chief expert
In: Op and St domain models; Key capabilities, operational requirements, measurements
Out: Selection criteria with assigned weights
Occurrence: Only once

Perform the Base Check

The base check refers to the initial analysis of SoS solution architectures. This check is intentionally separated from the evaluation of alternative solution architectures by the selection criteria. There are several reasons for carrying out a separate initial check: (1) determining the quality for each alternative solution architecture; (2) the set of solution architectures is narrowed down by choosing only the plausible one(s); (3) the final results of an evaluated alternative solution architecture may be distorted due to its poor quality.

In this stage, the trade study experts group is responsible for the initial checking of the architecture of each solution. This stage determines whether the alternative solution architecture is of sufficient quality for comprehensive evaluation by the defined trade study selection criteria.

Responsible role: Trade study experts group
In: Op and St domain models; Rs domain model; Key capabilities, operational requirements, measurements
Out: Base Check results on each alternative; Set of plausible alternatives
Occurrence: For each alternative solution architecture
Includes simulation: Yes

Perform Deep Check

The base check stage is completed by establishing a set of plausible alternative solution architectures. The deep check refers to the detailed analysis of the plausible alternative solution architectures by using the selection criteria. In this stage, the trade study experts group is responsible for checking that each plausible solution architecture is evaluated on the basis of the selection criteria.

Responsible role: Trade study experts group
In: Op and St domain models; Rs domain model; Selection criteria; Plausible alternative solution architectures
Out: Deep Check results on each alternative; Preferred alternative
Occurrence: For each alternative solution architecture
Includes simulation: Yes

Capture key decision and rationale

When the Deep Check has been completed, the decision-making authority reviews the Base Check and the Deep Check results in order to make a final decision. Any part of the trade study can be repeated if necessary to obtain more accurate data.

In this step, the decision-making authority captures the key decision in terms of whichever alternative solution architecture is the most balanced one based on the requirements. The decision shall be accompanied by a formal rationale. However, if none of the alternatives meets the requirements, the decision-making authority may refuse to select the preferred alternative solution. This step may lead to a review of the requirements and the evaluation criteria of the trade study.

Responsible role: Decision-Making Authority
In: Results of the Base Check and Deep Check
Out: Formal decision and rationale
Occurrence: Only once

Provide feedback for contractor

Once the decision-making authority has chosen the preferred alternative solution architecture, the end of the competition in the acquisition is announced.

In this step, the trade study leader or any other involved person provides the feedback of the trade study results to each contractor who has submitted their solution architecture.

Responsible role: Trade Study Lead
In: Formal decision and rationale
Out: Feedback of the trade study results to each contractor
Occurrence: For each alternative solution architecture

2.1.1 Base Check

The initial evaluation of the submitted SoS solution architectures begins with the Base Check (Fig. 2.4). The purpose of this stage is to check whether SoS solution architectures meet the key requirements and to determine their quality. Poor-quality architecture can be eliminated from the trade study. The following is a detailed explanation of all the stages of the Base Check.

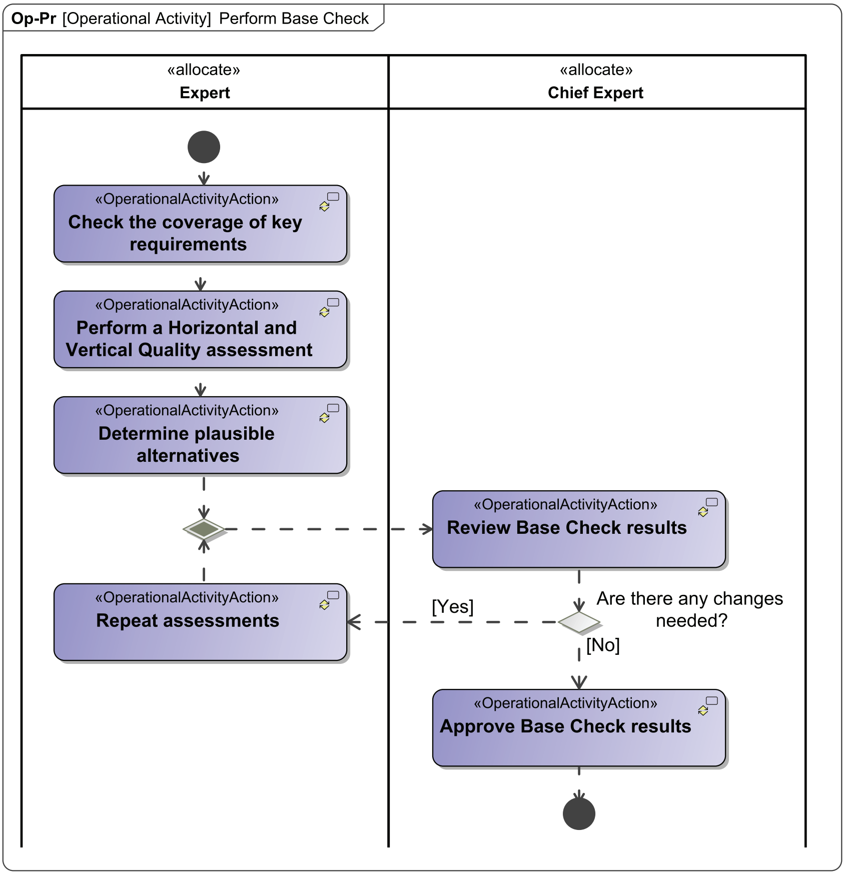

Op-Pr [Operational Activity] Perform Base Check
«allocate»
Expert
«allocate»
Chief Expert
«OperationalActivityAction»
Check the coverage of key requirements
«OperationalActivityAction»
Perform a Horizontal and Vertical Quality assessment
«OperationalActivityAction»
Determine plausible alternatives
«OperationalActivityAction»
Review Base Check results
«OperationalActivityAction»
Repeat assessments
[Yes]
Are there any changes needed?
[No]
«OperationalActivityAction»
Approve Base Check results

Fig. 2.4. UAF-based trade study subprocess – Base Check

Check the coverage of key requirements

The base check stage starts with the coverage check of key requirements that were set out in the 'Identified key requirements' stage. Satisfaction with the key capabilities, operational requirements and measurements implies that the required functionalities and features of the preferred solution are covered.

In this stage, the trade study expert is responsible for verifying that each submitted solution architecture (Rs domain model) meets the critical capabilities, operational requirements, and measurements. This stage should be followed by a formal inspection report containing details of the coverage of the key requirements.

Responsible role: Expert
In: St and Op domain models; Key capabilities, operational requirements, measurements; Plausible alternative solution architectures
Out: Report of the key requirements coverage
Occurrence: For each alternative solution architecture

Includes simulation: Yes

Perform a Horizontal and Vertical quality assessment

The horizontal quality assessment refers only to determining the quality of the solution architecture (Rs domain model). Vertical quality assessment refers to determining the quality of the solution architecture (Rs domain model) by checking traceability links against the St and Op domain models. This stage aims to determine the quality of each solution architecture so that only plausible solution architectures for the next stage would be selected.

In this stage, the trade study expert is responsible for determining the quality of each alternative solution architecture. The alternative Rs domain models are checked against the predefined St, Op, and Rs domain rules based on the UAF principles. This stage should be followed by a formal inspection report containing details of the quality assessment.

Responsible role: Trade Study Expert
In: St and Op domain models; Rs domain model; Horizontal and Vertical quality assessment rule set
Out: Quality assessment report
Occurrence: For each alternative solution architecture
Includes simulation: Yes

Determine plausible alternatives

The alternative solution architecture that has reached the acceptable quality level is called a plausible alternative.

In this stage, the trade study expert is responsible for determining a set of plausible solution architectures that will be evaluated in the subsequent stages. However, at this stage, a decision may be made to allow the contractors to refine their solution architecture so that their solution can be further evaluated.

Responsible role: Expert
In: Quality index for each alternative solution architecture
Out: Set of plausible alternatives

Occurrence: For each alternative solution architecture

Review Base Check results

In this stage, the chief expert reviews the results of the base check and the set of the plausible solution architectures. If necessary, any stage of the base check can be repeated.

Responsible role: Chief Expert
Inputs: Quality index for each alternative; Quality assessment report; Key requirements coverage report; Set of plausible alternatives
Occurrence: For each alternative solution architecture

Repeat assessments

In this stage, the trade study expert is responsible for repeating any required assessments of the plausible alternative solution architectures requested by the chief expert after reviewing the final results of the base check.

Responsible role: Expert
Inputs: St and Op domain models; Key capabilities, operational requirements, measurements; Plausible alternative solution architectures; Rs domain model; Horizontal and Vertical quality assessment rule set
Outputs: Quality assessment report; Key requirements coverage report; Set of plausible alternatives
Occurrence: For each alternative solution architecture

Approve Base Check results

This stage is followed by a formal decision and the rationale according to the established set of plausible solution architectures.

Responsible role: Chief Expert
Inputs: Quality index for each alternative; Quality assessment report; Key requirements coverage report; Set of plausible alternatives
Outputs: Official decision and rationale
Occurrence: For each alternative solution architecture

2.1.2 Deep Check

The comprehensive evaluation of the plausible SoS solution architectures begins with the Deep Check (Fig. 2.5). This stage aims to choose the most preferred solution architecture which meets the major requirements and characteristics. The following is a detailed explanation of all the stages of the Deep Check.

Evaluate each alternative by selection criteria

The deep check stage begins with the evaluation of each plausible solution architecture according to the established selection criteria. The plausible solution architectures are assessed by their general satisfaction with the desired characteristics.

In this stage, the trade study expert is responsible for evaluating each plausible solution architecture against the established selection criteria. Each evaluated plausible solution architecture receives its overall score.

Responsible role: Expert
In: Set of plausible solution architectures; Selection criteria

Out: Final score for each alternative; Analysis report
Occurrence: For each alternative solution architecture
Includes simulation: Yes

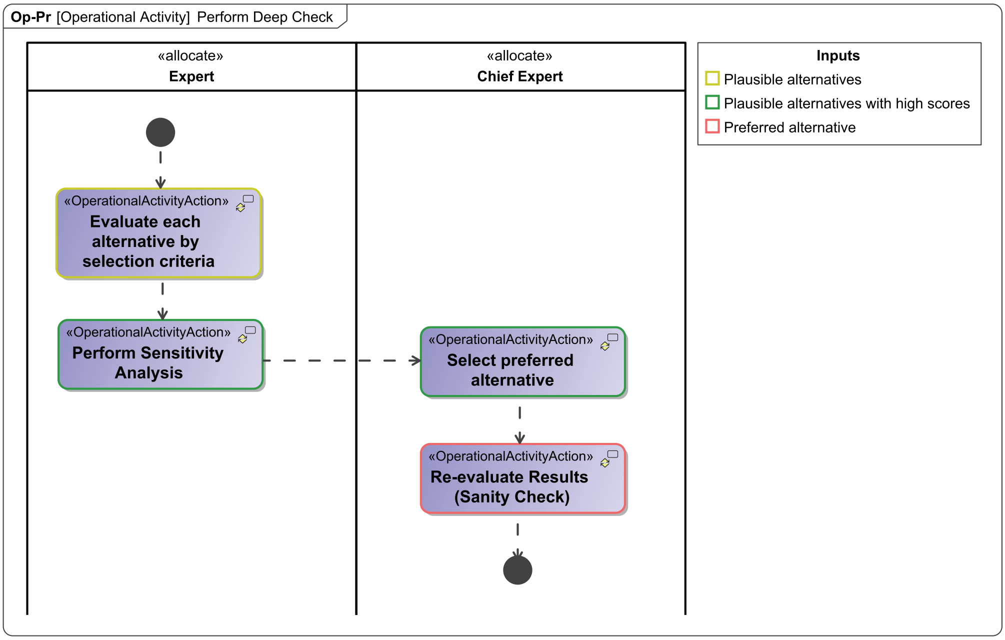

Op-Pr [Operational Activity] Perform Deep Check
«allocate»
Expert
«allocate»
Chief Expert
Inputs
Plausible alternatives
Plausible alternatives with high scores
Preferred alternative
«OperationalActivityAction»
Evaluate each alternative by selection criteria
«OperationalActivityAction»
Perform Sensitivity Analysis
«OperationalActivityAction»
Select preferred alternative
«OperationalActivityAction»
Re-evaluate Results (Sanity Check)

Fig. 2.5. UAF-based trade study subprocess – Deep Check

Perform sensitivity analysis

A summary of the evaluation results by the selection criteria may show that several alternatives have been given a close or even equal score. In this case, a sensitivity analysis is carried out to verify the choice of the preferred solution architecture. Since weight factors and quantitative data can have an arbitrary aspect, sensitivity analysis is essential [133].

In this stage, the trade study expert is responsible for performing the sensitivity analysis for the solution architectures getting similar scores. This stage may lead to a review of the selection criteria and their weight. The criteria that have no effect may be excluded from the scope of the sensitivity analysis.

Responsible role: Expert
In: Set of plausible solution architectures with a similar score
Out: Preferred alternative; Official report on the analysis results
Occurrence: For each alternative solution architecture with a similar score
Includes simulation: Yes

Select preferred solution architecture

In this stage, the trade study chief expert reviews the results of the deep check. This stage aims to choose the most preferred solution architecture based on the analysis results.

Responsible role: Chief Expert

In: Official reports on the base and deep analysis results
Out: Preferred alternative
Occurrence: Once

Sanity Check

Sanity check is an engineering judgment to ensure that the proper decision has been made. In this stage, the trade study chief expert is responsible for re-evaluating the chosen and preferred solution architecture. This stage aims to briefly confirm whether the preferred solution architecture satisfies the essential requirements and desirable features.

Responsible role: Chief Expert
In: Preferred solution architecture; Key capabilities, operational requirements, measurements; Report on the base and deep analysis results
Out: Report on the analysis results
Occurrence: Once

2.2 UAF-based trade study subprocess – Base check

This section features two parts. The first part describes a coverage check of the key capabilities, operational requirements, and measurements. The second part introduces a set of validation rules, which is defined with the objective to determine the quality of the architecture of each alternative solution.

2.2.1 Coverage check of key requirements

Satisfaction with the key requirements implies that the expected functionality of the preferred solution has been covered. In order to identify the key capabilities, operational requirements or measurements, the *Critical* stereotype is introduced (Fig. 2.6) that extends the UAF domain metamodel (DMM).

The coverage of the key capabilities and operational requirements is carried out by checking traceability links. The logical entity identified at the upper abstraction level must be associated with resources at the lower abstraction level.

As quantitative key measurements can be parametrized, a new verification process is introduced for that kind of verification (Fig. 2.7). Two types of requirements are checked in the course of the verification: parametric and dynamic. The parametric requirement defines a restriction only to a system parameter (e.g. the mass shall not be greater than 1500 kg). The dynamic requirement requires verification of a defined parameter at the specific behavioral moment of SoS (e.g. the wheel speed shall be equal or greater than the wheel speed before a gear upshift).

The following figure explains all the steps of the requirements verification process. A prerequisite is the identified key measurements.

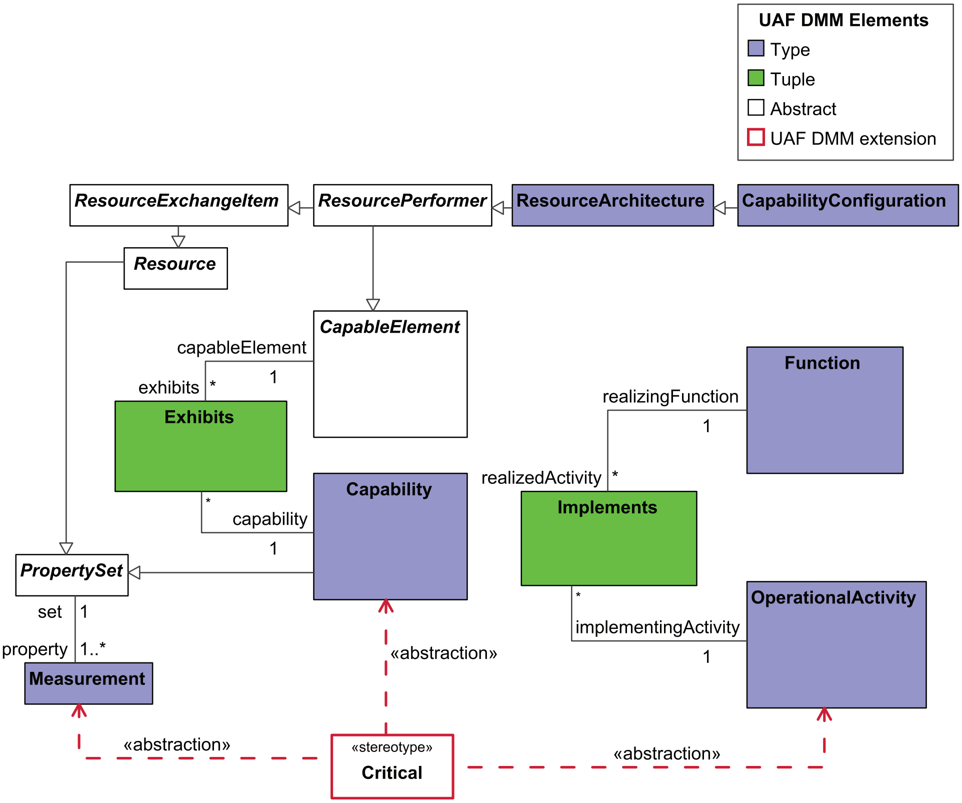


Fig. 2.6. Extended UAF DMM scope for the identification of key requirements

Define analysis context

In order to perform various analyses without modifying the original system architecture, the context of analysis should be used. "The analysis context reflects the structure of the analyzed system" [134]. In this step, the analysis context for the key requirements coverage should be defined.

Establish requirements

In order to automate the verification of requirements, the introduced key measurements should be linked to the *Requirement* element which reflects the necessary function, feature, or restriction. In this step, the *Requirement* element or any extended requirements (e.g. *functionalRequirement*, *performanceRequirement*, *interfaceRequirement*, etc.) should be established.

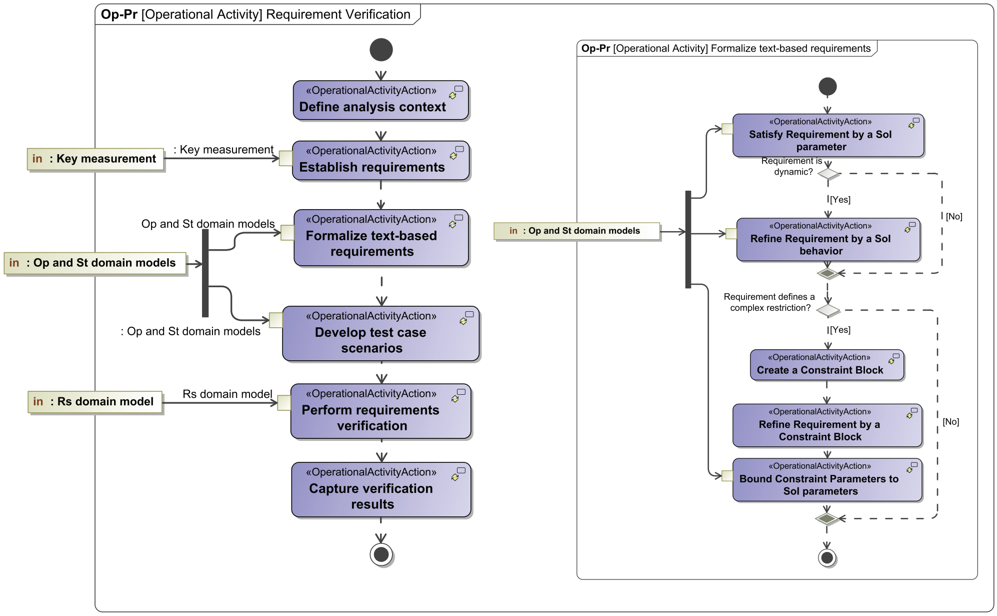

Op-Pr [Operational Activity] Requirement Verification
«OperationalActivityAction» Define analysis context
in : Key measurement
: Key measurement
«OperationalActivityAction» Establish requirements
Op and St domain models
in : Op and St domain models
«OperationalActivityAction» Formalize text-based requirements
: Op and St domain models
«OperationalActivityAction» Develop test case scenarios
in : Rs domain model
Rs domain model
«OperationalActivityAction» Perform requirements verification
«OperationalActivityAction» Capture verification results
Op-Pr [Operational Activity] Formalize text-based requirements
in : Op and St domain models
«OperationalActivityAction» Satisfy Requirement by a Sol parameter
Requirement is dynamic?
[Yes]
[No]
«OperationalActivityAction» Refine Requirement by a Sol behavior
Requirement defines a complex restriction?
[Yes]
[No]
«OperationalActivityAction» Create a Constraint Block
«OperationalActivityAction» Refine Requirement by a Constraint Block
«OperationalActivityAction» Bound Constraint Parameters to Sol parameters

Fig. 2.7. Requirements verification process for quantitative measurements

Formalize text-based requirements

SysML introduces text-based requirements, yet it does not specify the formalization rules. However, traceability links between the requirements and other elements provide essential information about the requirements. Source [71] stated that the *Refine* relationship should be used between the *Requirement* and the *Constraint Block* which can contain mathematical equations. The *Refine* relationship refers to specific and less abstract elements that allow to formalize the textual requirements.

The use of the *Refine* relationship in this verification process is expanded in order to formalize the dynamic requirements. *Requirement* is linked to a *State* or *Operational Activity* by indicating that a particular *Requirement* restriction only applies if the system is in a certain state or performs a certain activity.

In this step, traceability links must be defined between *Requirements* and specific model elements. First, *Requirement* should be linked to *Measurement* by using the *Satisfy* relationship. Second, if *Requirement* is dynamic, it should be related to the *OperationalActivity* or *State* with the *Refine* relationship. Third, if the *Requirement* defines a relatively complex constraint, a *Constraint Block* should be introduced and linked to the *Requirement* by using the *Refine* relationship. Fourth, if *Constraint Blocks* are used to formalize complex restrictions, the *Constraint Parameters* should be bounded to the *Measurements* that are verified by the *Requirement*. Fig. 2.8 shows the scope of UAF DMM for the formalization of the requirements.

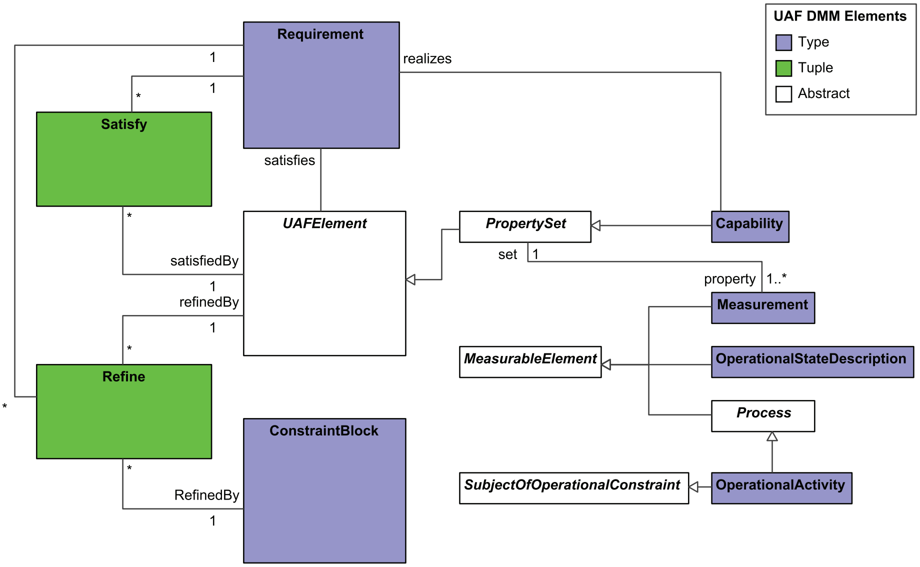


Fig. 2.8. UAF DMM scope for the formalization of requirements

Develop test case scenarios

Some system parameters vary depending on changes in the system behavior, such as the car speed, the stopping distance, or the order time. In order to verify these parameters in accordance with the specified requirements, a test case scenario should be designed by using the *Operational Process Flow* or *Operational Sequence* diagrams.

In this step, a *Test Case* scenario should be developed for the verification of dynamic requirements. A *Test Case* should be linked to a specific *Requirement* by using the *Verify* relationship. Fig. 2.9 shows the scope of UAF DMM for the development of a test case scenario.

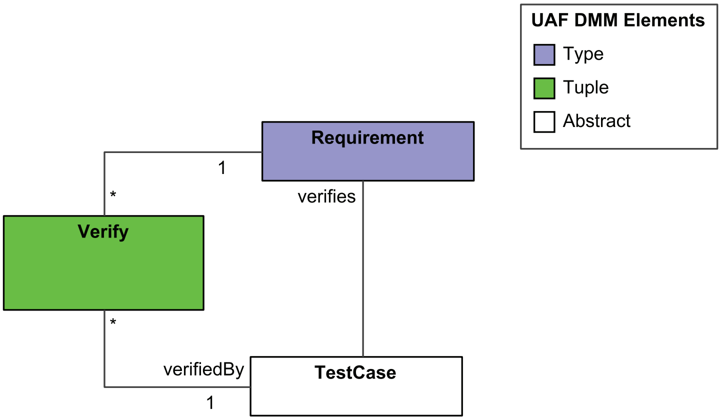


Fig. 2.9. UAF DMM scope for test case development

Perform requirements verification

The set of the submitted solution architectures is evaluated on the basis of their overall satisfaction with the established requirements. The requirement verification is performed according to the formalized *Requirements* and their traceability links.

The parametric requirement passes a verification when the verified parameter meets the specified restriction; otherwise, it fails. The dynamic requirement passes a verification when the SoS is in the required behavior, and when the verified parameter meets the specified restriction. The dynamic requirement fails a verification when the SoS is in the required behavior, but the verified parameter does not satisfy the specified restriction. The parametric requirement is marked as inconclusive when the verified parameter does not have a value. The dynamic requirement is marked as inconclusive when the verified parameter does not have a value or the SoS is in the invalid behavior. Table 2.1 provides an explanation for the verification of the parametric and dynamic requirements.

Table 2.1. Example of the verification of parametric and dynamic requirements

Requirement	Actual Value	Actual State	Parametric Verification	Dynamic Verification	Verification Status
Speed shall not exceed 220 km/h	210	-	Pass	-	Pass
	230	-	Fail	-	Fail
	N/A	-	Inconclusive	-	Inconclusive
Rescuing duration shall not exceed 28 hours	24	Rescuing	Pass	Pass	Pass
	30	Rescuing	Fail	Pass	Fail
	29	Idle	Fail	Inconclusive	Inconclusive
	20	Idle	Pass	Inconclusive	Inconclusive
	N/A	N/A	Inconclusive	Inconclusive	Inconclusive

Capture verification results

Once the verification of the key requirements has been completed, the results of each evaluated solution architecture should be recorded. Each verified requirement should have a status (fail/pass), a satisfied property, and a margin between the required and the actual values.

The process of verification of the introduced requirements is designed to check the satisfaction of the key requirements in order to indicate whether the alternative solution architecture is proper for a comprehensive trade study. However, the trade study lead may decide to skip this step and to analyze in detail all the selected alternative solution architectures.

2.2.2 Quality assessment

New sets of validation rules (Fig. 2.1.) are introduced to assess the quality of alternative solution architectures. In the scope of this thesis, quality assessment means verifying the completeness and correctness of SoS architecture models. The rule sets are adapted to be applied to automated methods during the base check.

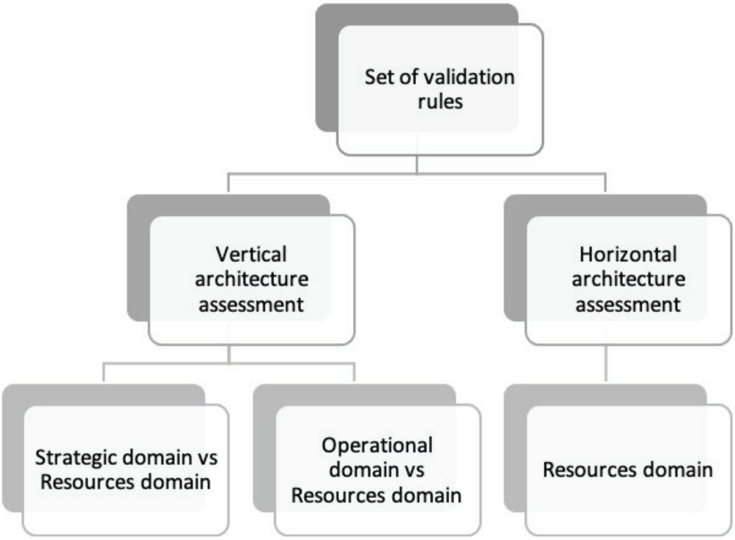

Set of validation rules
Vertical architecture assessment
Horizontal architecture assessment
Strategic domain vs Resources domain
Operational domain vs Resources domain
Resources domain

Fig. 2.10. Validation rules structure for architecture quality assessment

Each validation rule determines the constrained elements, the type (direct/direct reverse/derivative), the severity level, and the severity justification. Severity is marked with three levels: high (weight=3), medium (weight=2), and low (weight=1), depending on the importance and criticality of the rule. In this dissertation, the severity level is assigned to the rules based on the principles of the UAFP metamodel:

- High – this severity level is assigned to those rules that check the coverages of *Capabilities* and *Operational Activities*, as they are the key elements that ensure the core functions of a system. *Capabilities* and *Operational Activities* defined at the upper abstraction level must be associated with a semantically equivalent element at the lower abstraction level.
- Medium – this severity level is assigned to those rules that check the coverage of the Op domain elements (excluding *Operational Activity*) by the Rs domain elements. Those Op domain elements define the *Operational Activity,* and they are required to ensure the completeness of the system architecture. In addition, this severity level is assigned to a rule that checks the coverage of a *Function* in the point of view of the Rs domain, since the *Function* represents the fundamental functionality of the investigated system.
- Low – this severity level is assigned to those rules that check for redundant elements in the Rs domain models (excluding *Function*).

The results of the horizontal and vertical quality assessments are provided in the calculation of the quality index, Equation (4). “The ratio of violations to executed rules is a general method of comparing two amounts” [135].

$$Q = 1 - \left(\frac{\sum_{s=1}^{3} w_s \times vr_s}{\sum_{s=1}^{3} w_s \times er_s} \div 3\right); \tag{4}$$

where:
Q – quality index,
s – severity level,
w – weight,
vr_s – number of validation rules violations based on severity *s*,
er_s – number of validation rules based on severity *s*.

The minimum recommended SoS architecture quality threshold is 0.7 (70%); this threshold is suggested based on the MIPS and CMS studies [136]. Those SoS architectures that do not meet the minimum quality threshold should be excluded from the further trade study process.

This section includes two parts. The first part introduces a set of the validation rules designed to check the traceability links among Rs domain models, which called the horizontal quality assessment. The second part introduces a set of the validation rules designed to check the traceability links among three UAF domain views: Rs, St and Op, which is called the vertical quality assessment.

2.2.2.1 Horizontal Quality Assessment

The validation rules of horizontal quality assessment are designed to verify the submitted SoS solution architectures. This assessment checks only the Rs domain models and their traceability links. The Rs domain refers to the definition of the solution architectures for implementing the Op requirements [132].

The set of the horizontal quality rules includes two traceability rules which are strictly based on UAFP. There are two types of rules: (1) direct (from the Rs point of view); (2) direct reverse (from the *Function* point of view). The following part presents the predefined validation rules that are designed for the horizontal quality assessment.

Rule #1 – Resource Performer performs a Function

Fig. 2.11 shows the scope of UAF DMM for Rule #1, and Table 2.2 provides a detailed description of Rule #1.

Table 2.2. Rule #1 description

Validation Rule	Severity	Description
∀a[RsPerformer(a) ∧ Atomic(a) → ∃b[Function(b) ∧ Performs(a,b)]] – *RsPerformer(a) – a is an element stereotyped by the <<ResourcePerformer>>* – *Atomic(a) – a is an atomic element*	**Low** (weight=1)	**Rule**: Each atomic *Resource Performer* must perform at least one *Function* **Type**: Direct **Constrained Element**: *Resource Performer* **Weight Justification**: The *IsCapableToPerform* relationship between *Resource Performer* and *Function* confirms that the introduced

– *Function(b) – b is an element stereotyped by the <<Function>>* – *Performs(a,b) – a is capable to perform b*		*Resource* performs a specific function. A *Resource Performer*, which does not perform any *Function* is redundant and should be removed or associated with a specific *Function*

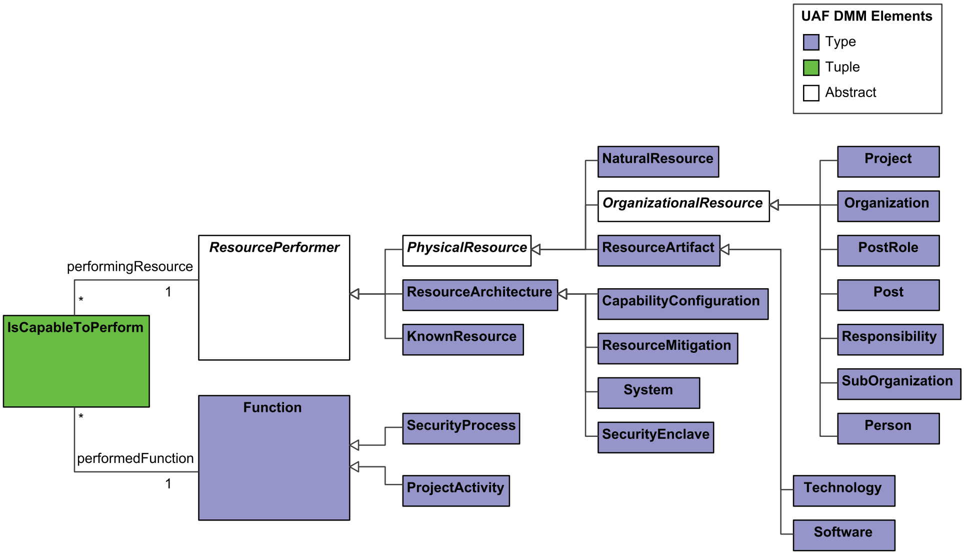


Fig. 2.11. UAF DMM scope for Rule #1 and Rule #2

Rule #2 – Function is performed by Resource Performer

Fig. 2.11 shows the scope of UAF DMM for Rule #2, and Table 2.3 provides a detailed description of Rule #2.

Table 2.3. Rule #2 description

Validation Rule	Severity	Description
∀a[Function(a) ∧ Atomic(a) → ∃b[RsPerformer(b) ∧ Performs(a,b)]] – *Function (a) – a is an element stereotyped by the <<Function>>* – *Atomic(a) – a is an atomic element* – *RsPerformer(b) – y is an element stereotyped by the <<ResourcePerformer>>* – *Performs(a,b) – b is capable to perform a*	**Medium** (weight=2)	**Rule**: Each atomic *Function* must be performed by a *Resource Performer* **Type**: Direct Reverse **Constrained Element**: *Function* **Weight Justification**: The *IsCapableToPerform* relationship between *Function* and *Resource Performer* confirms that the introduced *Function* is used in the architecture. A *Function* that no *Resource Performer* performs is redundant and should be removed or associated with a specific *Resource Performer*

2.2.2.2 Vertical Quality Assessment

The vertical quality assessment validation rules are designed to verify the submitted SoS solution architectures against the requirements. This assessment checks St, Op, and Rs domain models and their traceability links. The set of rules is divided into two categories based on the inspected domains: (1) St domain vs Rs domain; (2) Op domain vs Rs domain. The following part presents the predefined validation rules of these two categories that are designed for vertical quality assessment.

St domain vs Rs domain

Validation rules of St vs Rs vertical quality assessment are designed to verify the submitted solution architectures and the St domain model. This assessment checks the traceability links between the Rs domain model and the St domain model. The St domain model describes the enterprise's goals, its vision, and the set of the necessary capabilities to achieve the stated goals [132]. In addition, this domain introduces the traceability links between the capability requirements and the resources needed to realize them.

The St vs Rs quality rules set includes two traceability rules which are strictly based on UAFP. There are two types of rules: (1) direct (from the *Capability* point of view); and (2) direct reverse (from the *Capability Configuration* point of view). The following part presents the predefined validation rules that are designed for St vs Rs vertical quality assessment.

Rule #3 – Capability is exhibited by Resource Performer

Fig. 2.12 shows the scope of UAF DMM for Rule #3, and Table 2.4 provides a detailed description of Rule #3.

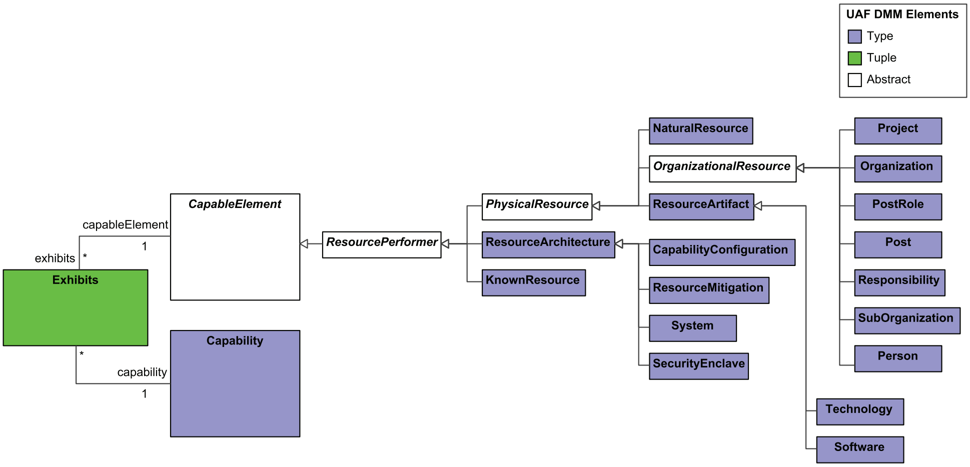


Fig. 2.12. UAF DMM scope for Rule #3 and Rule #4

Table 2.4. Rule #3 description

Validation Rule	Severity	Description
$\forall a[\text{Capability}(a) \wedge \text{Atomic}(a) \rightarrow \exists b[\text{RsPerformer}(b) \wedge \text{Exhibits}(b,a)]]$ – *Capability(a) – a is an element stereotyped by the <<Capability>>* – *Atomic(a) – a is an atomic element* – *RsPerformer(b) – b is an element stereotyped by the <<ResourcePerformer>>* – *Exhibits(b,a) – b exhibits a*	**High** (weight=3)	**Rule**: Each atomic *Capability* must be exhibited by a *Resource Performer* **Type**: Direct **Constrained Element**: *Capability* **Weight Justification**: The *Exhibits* relationship confirms that a *Capability* is carried out by a specific *Resource Performer*. A *Capability* that no *Resource Performer* exhibits is redundant and should be removed or associated with a specific *Resource Performer*

Rule #4 – Capability Configuration exhibits a Capability

Fig. 2.12 shows the scope of UAF DMM for Rule #4, and Table 2.5 provides a detailed description of Rule #4.

Table 2.5. Rule #4 description

Validation Rule	Severity	Description
$\forall a[\text{CapabilityConfig}(a) \wedge \text{Atomic}(a) \rightarrow \exists b[\text{Capability}(b) \wedge \text{Exhibits}(a,b)]]$ – *CapabilityConfig(a) – a is an element stereotyped by the <<CapabilityConfiguration>>* – *Atomic (a) – a is an atomic element* – *Capability(b) – b is an element stereotyped by the <<Capability>>* – *Exhibits(a,b) – a exhibits b*	**Low** (weight=1)	**Rule Description**: Each atomic *Capability Configuration* must exhibit at least one *Capability* **Type**: Direct Reverse **Constrained Element**: *Capability Configuration* **Weight Justification**: The composite structure reflects the physical and human resources that are gathered to meet the capabilities. The *Exhibits* relationship between *Capability Configuration* and *Capability* confirms that the introduced *Capability Configuration* is used in the architecture. A *Capability Configuration* which does not exhibit any *Capability* is redundant and should be removed or associated with a specific *Capability*

Op domain vs Rs domain

The validation rules of Op vs Rs vertical quality assessment are designed to verify the submitted SoS solution architectures and the provided logical architecture. This assessment checks the traceability links between the Rs domain model and the

Op domain model. The Op domain model describes the logical architecture and sets the operational requirements as well as the operational behaviors to support the specified capabilities. This domain defines operational elements regardless of implementation [132].

The Op vs Rs quality rules set includes ten traceability rules which are strictly based on UAFP. There are three types of rules: (1) direct (from the Op point of view); (2) direct reverse (from the Rs point of view); (3) derivative (a derived traceability link related to other relationships). The following part presents predefined validation rules that are designed for Op vs Rs vertical quality assessment.

Rule #5 – Operational Performer is implemented by a Resource Performer

Fig. 2.13 shows the scope of UAF DMM for Rule #5, and Table 2.6 provides a detailed description of Rule #5.

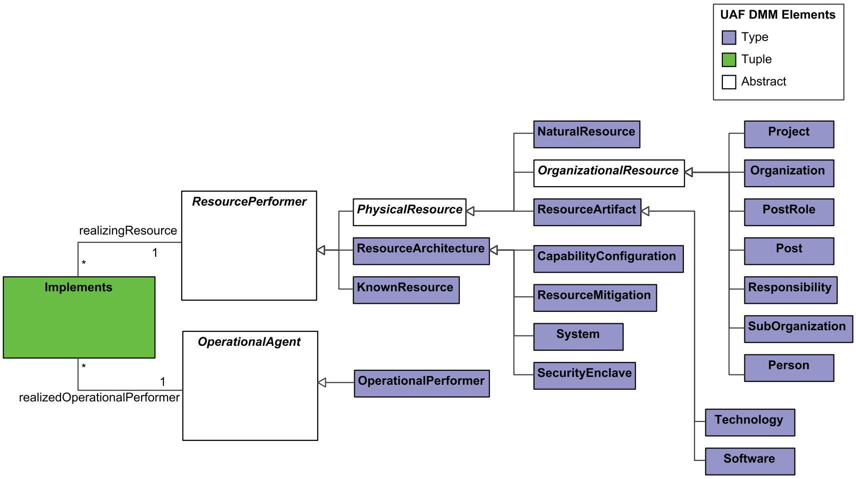


Fig. 2.13. UAF DMM scope for Rule #5 and Rule #6

Table 2.6. Rule #5 description

Validation Rule	Severity	Description
∀a[OpPerformer(a) ∧ Atomic(a) → ∃b[RsPerformer(b) ∧ Implements(b,a)]] – *OpPerformer(a) – a is an element stereotyped by the <<OperationalPerformer>>* – *Atomic (a) – a is an atomic element* – *RsPerformer(b) – b is an element stereotyped by the <<ResourcePerformer>>* – *Implements(b,a) – b implements a*	**Medium** (weight=2)	**Rule**: Each atomic *Operational Performer* must be implemented by a *Resource Performer* **Type**: Direct **Constrained Element**: *Operational Performer* **Weight Justification**: The logical entity identified at the upper abstraction layer must be associated with resources from the lower level of abstraction. The *Implements* relationship confirms that the introduced *Operational Performer* is

		implemented by a semantically equivalent element at the lower abstraction level. An *Operational Performer* that no *Resource Performer* implements is redundant and should be removed or associated with a specific *Resource Performer*

Rule #6 – Resource Performer implements an Operational Performer

Fig. 2.13 shows the scope of UAF DMM for Rule #6, and Table 2.7 provides a detailed description of Rule #6.

Table 2.7. Rule #6 description

Validation Rule	Severity	Description
∀a[RsPerformer(a) ∧ Atomic(a) → ∃b[OpPerformer(b) ∧ Implements(a,b)]] – *RsPerformer (a) – a is an element stereotyped by the <<OperationalExchange>>* – *Atomic (a) – a is an atomic element* – *OpPerformer (b) – b is an element stereotyped by the <<ResourceExchange>>* – *Implements(a,b) – a implements b*	**Low** (weight=1)	**Rule**: Each atomic *Resource Performer* must implement at least one *Operational Performer* **Type**: Direct Reverse **Constrained Element**: *Resource Performer* **Weight Justification**: The *Implements* relationship between the *Resource Performer* and the *Operational Performer* confirms that the introduced *Resource Performer* at the lower abstraction level is used in the architecture. A *Resource Performer* which does not implement any *Operational Performer* is redundant and should be removed or associated with a specific *Operational Performer*

Rule #7 – Operational Activity is implemented by a Function

Fig. 2.14 shows the scope of UAF DMM for Rule #7, and Table 2.8 provides a detailed description of Rule #7.

Table 2.8. Rule #7 description

Validation Rule	Severity	Description
∀a[OpActivity(a) ∧ Atomic(a) → ∃b[Function(b) ∧ Implements(b,a)]] – *OpActivity(a) – a is an element stereotyped by the <<OperationalActivity>>* – *Atomic (a) – a is an atomic element*	**High** (weight=3)	**Rule**: Each atomic *Operational Activity* must be implemented by a *Function* **Type**: Direct **Constrained Element**: *Operational Activity* **Weight Justification**: The *Operational Activities* identified in the upper abstraction layer must be associated with *Functions* from the lower level of

– *Function(b) – b is an element stereotyped by the <<Function>>* – *Implements(b,a) – b implements a*		abstraction. The *Implements* relationship confirms that the introduced *Operational Activity* is implemented by a semantically equivalent element at the lower abstraction level. An *Operational Activity* that no *Function* implements is redundant and should be removed or associated with a specific *Function*

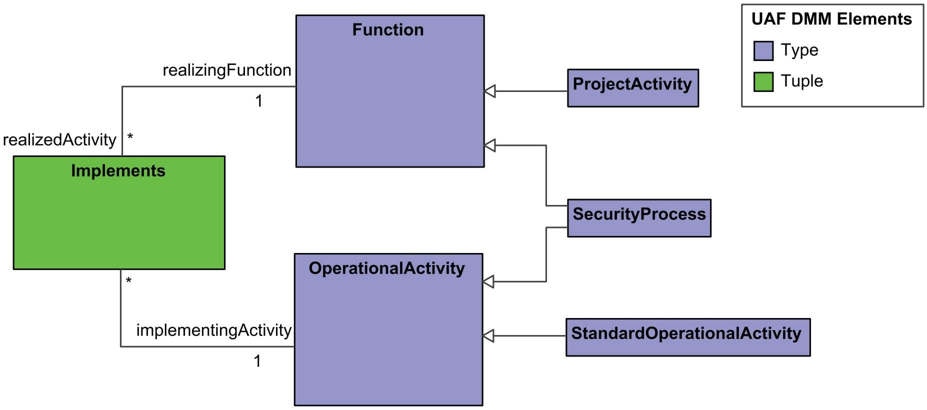


Fig. 2.14. UAF DMM scope for Rule #7 and Rule #8

Rule #8 – Function implements an Operational Activity

Fig. 2.14 shows the scope of UAF DMM for Rule #8, and Table 2.9 provides a detailed description of Rule #8.

Table 2.9. Rule #8 description

Validation Rule	Severity	Description
∀a[Function(a) ∧ Atomic(a) → ∃b[OpActivity(b) ∧ Implements(a,b)]] – *Function (a) – a is an element stereotyped by the <<Function>>* – *Atomic (a) – a is an atomic element* – *OpActivity(b) – b is an element stereotyped by the <<OperationalActivity>>* – *Implements(a,b) – a implements b*	**Low** (weight=1)	**Rule**: Each atomic *Function* must implement at least one *Operational Activity* **Type**: Direct Reverse **Constrained Element**: *Function* **Weight Justification**: the *Functions* specified in the context of the *Resource Performer* reflect *Operational Activities* in the upper abstraction layer. The *Implements* relationship between the *Function* and the *Operational Activity* confirms that the introduced *Function* performs the actions of the enterprise's business activities. A *Function* that does not implement any *Operational Activity* is redundant and should be removed or

		associated with a specific *Operational Activity*

Rule #9 – Information Element is implemented by a Data Element

Fig. 2.15 shows the scope of UAF DMM for Rule #9, and Table 2.10 provides a detailed description of Rule #9.

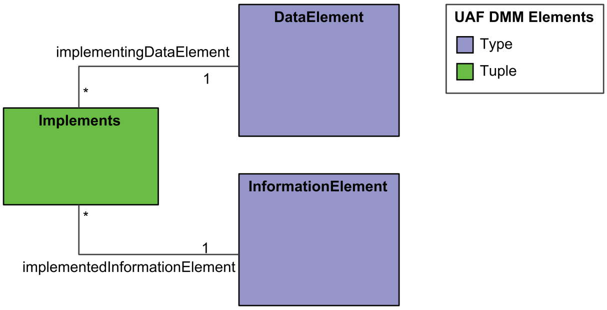


Fig. 2.15. UAF DMM scope for Rule #9 and Rule #10

Table 2.10. Rule #9 description

Validation Rule	Severity	Description
∀a[InfoElement(a) → ∃b[DataElement(b) ∧ Implements(b,a)]] – *InfoElement (a) – a is an element stereotyped by the <<InformationElement>>* – *DataElement (b) – b is an element stereotyped by the <<DataElement>>* – *Implements(b,a) – b implements a*	**Medium** (weight=2)	**Rule**: *Information Element* must be implemented by a *Data Element* **Type**: Direct **Constrained Element**: *Information Element* **Weight Justification**: The *Information Elements* identified in the upper abstraction layer must be associated with the *Data Elements* from the lower level of abstraction. The *Implements* relationship confirms that the introduced *Information Element* is implemented by a semantically equivalent element at the lower abstraction level. An *Information Element* which no *Data Element* implements is redundant and should be removed or associated with a specific *Data Element*

Rule #10 – Data Element implements an Information Element

Fig. 2.15 shows the scope of UAF DMM for Rule #10, and Table 2.11 provides a detailed description of Rule #10.

Table 2.11. Rule #10 description

Validation Rule	Severity	Description
∀a[DataElement (a) → ∃b[InfoElement(b) ∧ Implements(a,b)]] – *DataElement (a) – a is an element stereotyped by the <<DataElement>>* – *InfoElement (b) – b is an element stereotyped by the <<InformationElement>>* – *Implements(a,b) – a implements b*	**Low** (weight=1)	**Rule Description**: *Data Element* must implement at least one *Information Element* **Type**: Direct Reverse **Constrained Element**: *Data Element* **Weight Justification**: *Data Elements* reflect *Information Elements* in the upper abstraction layer. The *Implements* relationship between the *Data Element* and the *Information Element* confirms that the introduced *Data Element* at the lower abstraction level is used in the architecture. A *Data Element* which does not implement any *Information Element* is redundant and should be removed or associated with a specific *Information Element*

Rule #11 – Operational Exchange is implemented by a Resource Exchange

Fig. 2.16 shows the scope of UAF DMM for Rule #11, and Table 2.12 provides a detailed description of Rule #11.

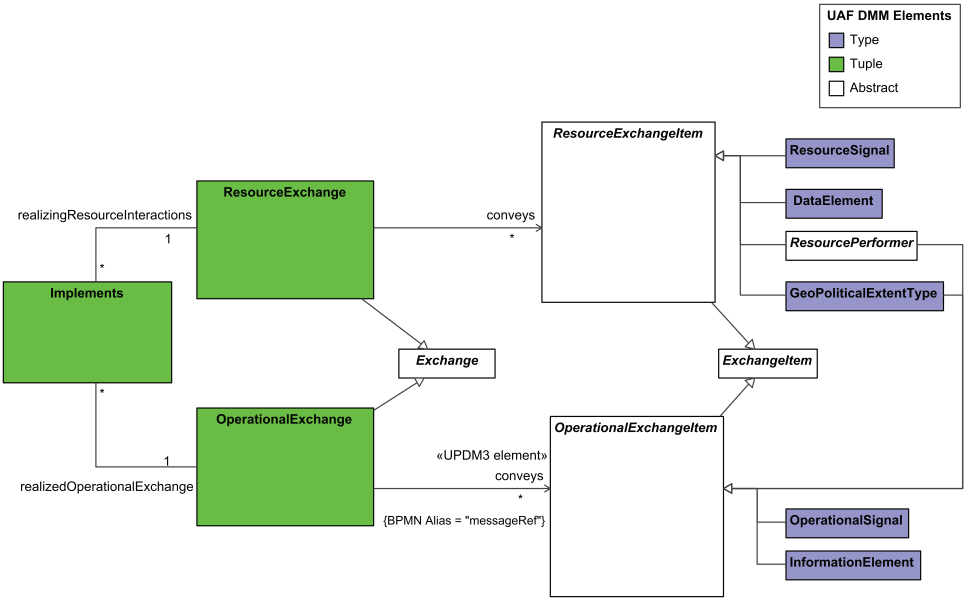


Fig. 2.16. UAF DMM scope for Rule #11 and Rule #12

Table 2.12. Rule #11 description

Validation Rule	Severity	Description
∀a[OpExchange(a) → ∃b[RsExchange(b) ∧ Implements(b,a)]] – *OpExchange (a) – a is an element stereotyped by the <<OperationalExchange>>* – *RsExchange (b) – b is an element stereotyped by the <<ResourceExchange>>* – *Implements(b,a) – b implements a*	**Medium** (weight=2)	**Rule Description**: *Operational Exchange* must be implemented by *Resource Exchange* **Type**: Direct **Constrained Element**: *Operational Exchange* **Weight Justification**: The *Operational Exchanges* identified in the upper abstraction layer must be associated with *Resource Exchanges* from the lower abstraction level. The *Implements* relationship confirms that the introduced *Operational Exchange* is implemented by a semantically equivalent element at the lower abstraction level. An *Operational Exchange* that no *Resource Exchange* implements is redundant and should be removed or associated with a specific *Resource Exchange*

Rule #12 – Resource Exchange implements an Operational Exchange

Fig. 2.16 shows the scope of UAF DMM for Rule #12, and Table 2.13 provides a detailed description of Rule #12.

Table 2.13. Rule #12 description

Validation Rule	Severity	Description
∀a[RsExchange (a) → ∃b[OpExchange(b) ∧ Implements(a,b)]] – *RsExchange (a) – a is an element stereotyped by the <<ResourceExchange>>* – *OpExchange (b) – b is an element stereotyped by the <<OperationalExchange>>* – *Implements(a,b) – a implements b*	**Low** (weight=1)	**Rule**: *Resource Exchange* must implement at least one *Operational Exchange* **Type**: Direct Reverse **Constrained Element**: *Resource Exchange* **Weight Justification**: *Resource Exchanges* reflect *Operational Exchanges* in the upper abstraction layer. The *Implements* relationship between the *Resource Exchange* and the *Operational Exchange* confirms that the introduced *Resource Exchange* at the lower abstraction level is used in the architecture. A *Resource* Exchange which does not implement any *Operational Exchange* is redundant and

		should be removed or associated with a specific *Operational Exchange*

Rule #13 – Resource Performer implements an Operational Performer and exhibits the same Capability as the implemented Operational Performer

Fig. 2.17 shows the scope of UAF DMM for Rule #13, and Table 2.14 provides a detailed description of Rule #13.

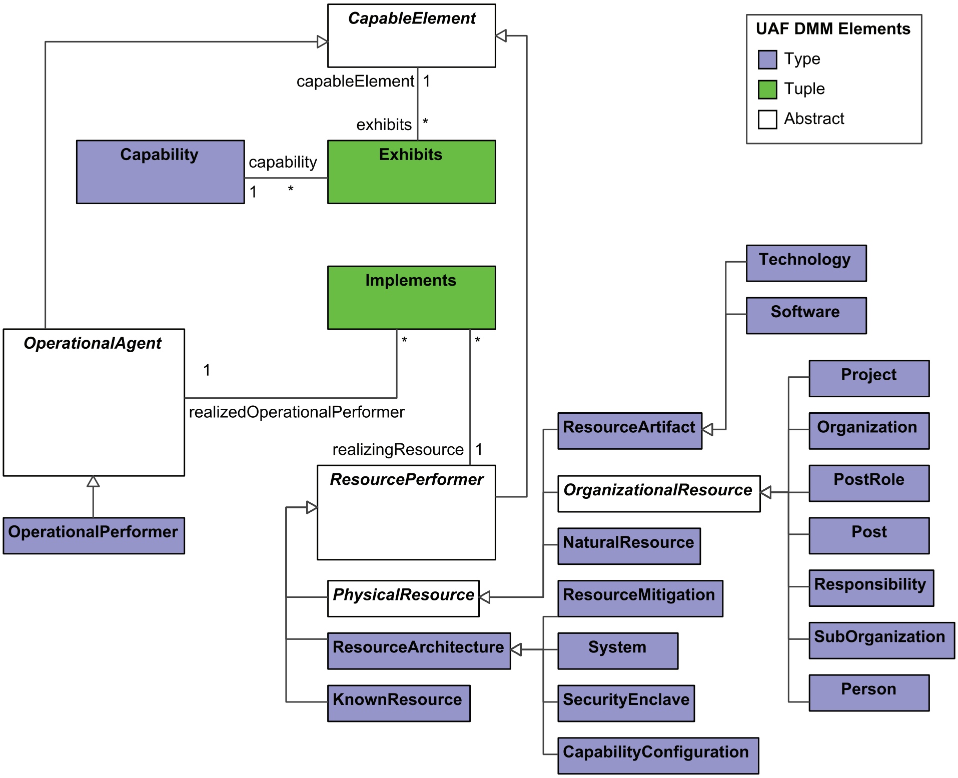


Fig. 2.17. UAF DMM scope for Rule #13

Table 2.14. Rule #13 description

Validation Rule	Severity	Description
$\forall a[(\text{RsPerformer}(a) \land \text{Atomic}(a)) \rightarrow$ $\exists b[\text{OpPerformer}(b) \land$ $\text{Implements}(a,b)]] \rightarrow$ $\exists c[\text{Capability}(c) \land \text{Exhibits}(b,c)]$ $\rightarrow[\text{Exhibits}(a,c)]$ – *RsPerformer (a) – a is an element stereotyped by the <<ResourcePerformer>>*	**Medium** (weight=2)	**Rule**: When an atomic *Resource Performer* implements an *Operational Performer*, then that *Resource Performer* must exhibit the same *Capability* as the implemented *Operational Performer* **Type**: Derivative **Constrained Element**: *Resource Performer*

– *Atomic (a) – a is an atomic element* – *OpPerformer (b) – b is an element stereotyped by the <<OperationalPerformer>>* – *Implements(a,b) – a implements b* – *Capability(a) – a is an element stereotyped by the <<Capability>>* – *Exhibits(b,c) – b exhibits c* – *Exhibits (a,c) – a exhibits c*		**Weight Justification**: The *Implements* relationship between the *Resource Performer* and the *Operational Performer* confirms that the *Operational Performer* is implemented by a semantically equivalent element at the lower abstraction level. In addition, the associated *Resource Performer* and the *Operational Performer* shall exhibit the same *Capability* as a *Resource Performer* reflects *Operational Performer* in the lower abstraction layer

Rule #14 – Function implements an Operational Activity and maps the same Capability as the implemented Operational Activity

Fig. 2.18 shows the scope of UAF DMM for Rule #14, and Table 2.15 provides a detailed description of Rule #14.

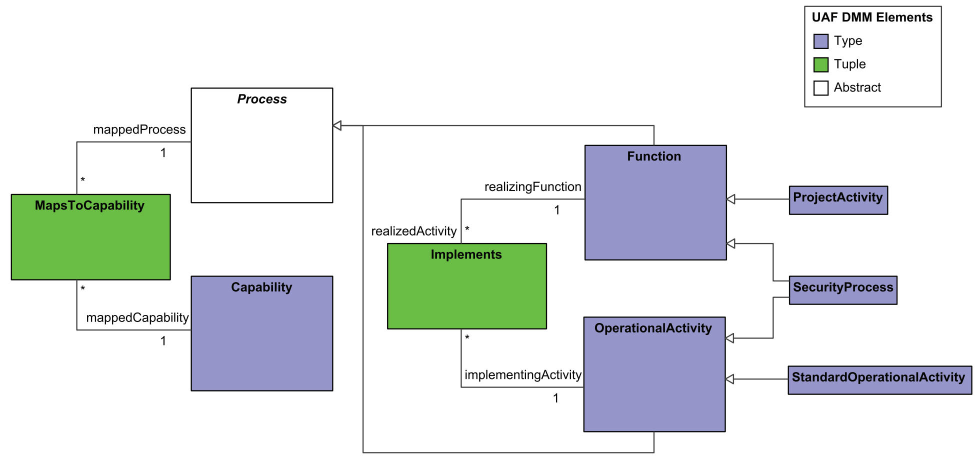


Fig. 2.18. UAF DMM scope for Rule #14

Table 2.15. Rule #14 description

Validation Rule	Severity	Description
$\forall a[(\text{Function}(a) \land \text{Atomic}(a)) \rightarrow \exists b[\text{OpActivity}(b) \land \text{Implements}(a,b)]] \rightarrow \exists c[\text{Capability}(c) \land \text{Maps}(b,c)] \rightarrow [\text{Maps}(a,c)]$ – *Function (a) – a is an element with a stereotype <<Function>>*	**Medium** (weight=2)	**Rule**: When an atomic *Function* implements an *Operational Activity*, then that *Function* must map the same *Capability* as the implemented Operational *Activity* **Type**: Derivative **Constrained Element**: *Function*

– *Atomic (a) – a is* an atomic element – *OpActivity (b) – b is an element stereotyped by the <<OperationalActivity>>* – *Implements(a,b) – a implements b* – *Capability(a) – a is an element stereotyped by the <<Capability>>* – *Maps(b,c) – b maps to capability c* – *Maps(a,c) – a maps to capability c*	**Weight Justification**: The *Implements* relationship between the *Function* and the *Operational Activity* confirms that the *Operational Activity* is implemented by a semantically equivalent element at the lower abstraction level. In addition, the associated *Function* and the *Operational Activity* shall map the same *Capability* as a *Function* reflects an *Operational Activity* in the lower abstraction layer

The quality assessment has introduced fourteen predefined validation rules that check the alternative solution architectures by evaluating the traceability links between three UAF domains: St, Op, and Rs. Each of the introduced rules has a predefined severity level that can be modified. Additionally, the quality assessment may be supplemented by more validation rules if necessary.

2.3 UAF-based trade study subprocess – Deep check

This section contains three parts. The first part introduces the initial data required to perform an evaluation by using the selection criteria. The second part presents the MBSE-based architecture evaluation algorithm by using the selection criteria. The third part describes the MBSE-based sensitivity analysis algorithm.

2.3.1 Initial data set

The deep check stage begins with an evaluation of the architectures of each plausible solution based on the established set of the trade study selection criteria. The plausible solution architectures are evaluated by their satisfaction with the desired characteristics.

In order to start the evaluation of the plausible solution architectures, the set of the required data should be gathered. The following list provides the set of the necessary data (Fig. 2.19):

- Selection criteria
- Selection criteria priority level
- Alternative solution architectures

The following part presents the data required to perform the evaluation of the plausible solution architectures.

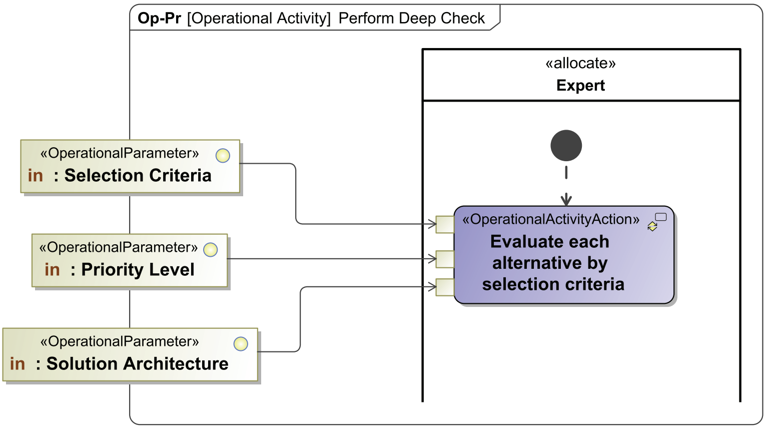

Op-Pr [Operational Activity] Perform Deep Check
«allocate»
Expert
«OperationalParameter»
in : Selection Criteria
«OperationalParameter»
in : Priority Level
«OperationalParameter»
in : Solution Architecture
«OperationalActivityAction»
Evaluate each alternative by selection criteria

Fig. 2.19. Initial data set for deep check

Selection criteria

A set of trade study selection criteria is established to evaluate alternative solution architectures. The selection criteria can be quantitative or qualitative, which can be specified by text. However, for automatic evaluation purposes, text-based selection criteria should be converted into *measures of effectiveness* (moe). "Moes are measures designed to reflect the achievement of mission objectives and the achievement of desired results" [60]. Moes are represented by a numeric expression, which allows them to be used to assess the compliance of moes with the outlined goals.

From the point of view of UAF DMM

The *Requirement* element is used to establish a set of text-based selection criteria. The specified R*equirement* element is linked with the *Capability* by the *Refine* relationship. In addition, the *PrioritizedRequirement* stereotype is applied for the *requirements* to the indicated *requirements* as the selection criteria for a trade study.

In order to automate the evaluation of architectures, a *CapabilityConfiguration* element is used, which corresponds to a specific *Capability* using the *Exhibits* relationship. A set of *Measurements* is introduced for a *CapabilityConfiguration* that reflects the selection criteria of the trade study. Each *Measurement* is associated with a specific *Requirement* through the *Satisfy* relationship and has the *Criterion* stereotype applied to it.

The *PrioritizedRequirement* and *Criterion* stereotypes are newly introduced, which extends UAF DMM to specify the selection criteria directly in the UAF model (Fig. 2.20).

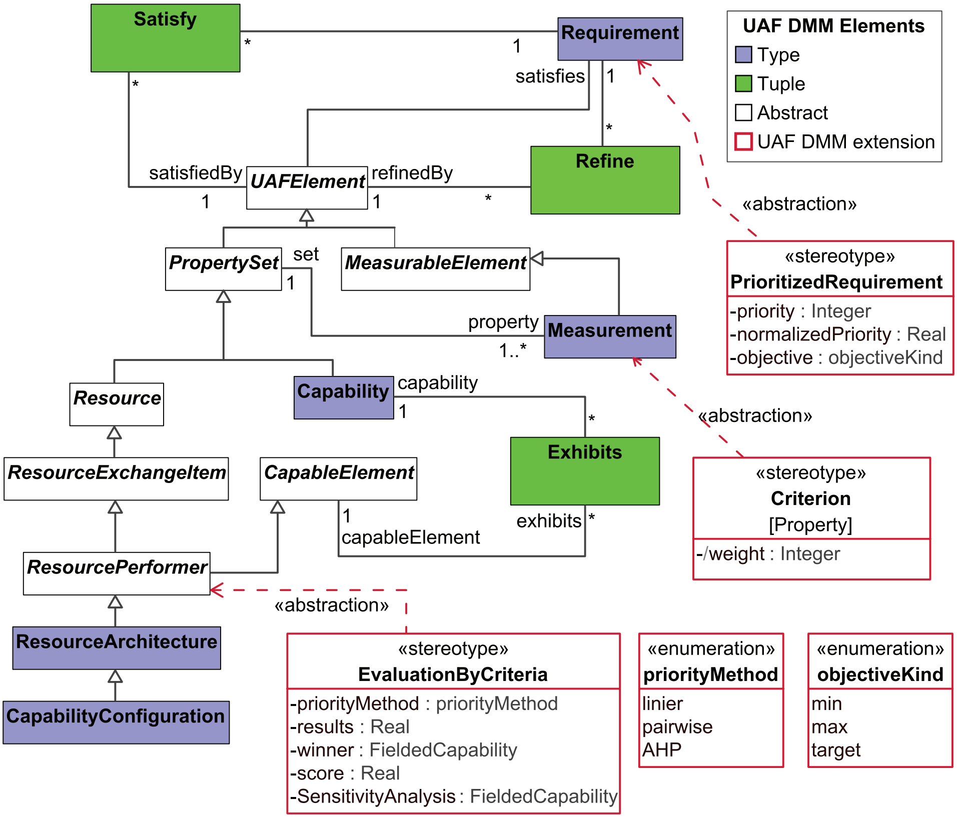


Fig. 2.20. UAF DMM extension for defining selection criteria

Priority level

Different selection criteria indicate a different level of importance. Priorities are assigned to indicate the importance level. The higher is the priority, the stronger the criterion affects the overall score. UT3SA is designed to support three ways to specify the priorities: linear, pairwise, and AHP.

Linear

For each selection criterion, the priority is indicated depending on its level of importance. To specify the priority, a predefined scale is used which is aligned with the trade study lead (e.g. one to three, where 'one' indicates a low level of importance; 'two' means a medium level of importance, and 'three' represents a high level of importance). Then, the priorities determined for the selection criteria are normalized by using Equation (5).

$$np_i = \frac{p_i}{\sum p_i}; \qquad (5)$$

Where:

np_i – normalized priority value of the i^{th} selection criterion,
p_i – i^{th} priority value,
Σp_i – sum of all priorities assigned to the selection criterion.

Pairwise

"Pairwise comparison compares alternatives in pairs to assess which of the alternatives is the most suitable" [62]. This method uses a pairwise matrix that indicates which alternative is more important than the others. Furthermore, the level of importance could be determined on the basis of a more precise level of importance (Table 2.16).

Table 2.16. Example scale for comparison [137]

Value	**Description**
1	Equal importance
3	Moderate importance
5	Strong or essential importance
7	Very strong importance
9	Extreme importance
2, 4, 6, 8	Intermediate values

After determining the level of importance, the geometric mean of each row of the matrix is calculated by using Equation (6). Finally, the calculated values are normalized by using Equation (5).

$$\mu GM_i = \sqrt[n]{\prod_i^n x_i}\,; \tag{6}$$

Where:

μGM_i – geometric mean of the ith row of pairwise matrix,
n – number of criteria in a row,
x_i – ith criterion value.

AHP

"AHP is a multi-attribute method that provides means of determining and selecting weight criteria" [104]. This method uses a pairwise matrix that indicates which of the alternatives is more important. After determining the level of importance, each record in the column is divided by the total sum of the column values to get a normalized result. Finally, each row of normalized results based on the column is summed up and normalized by using Equation (5).

From the point of view of UAF DMM

The *PrioritizedRequirement* stereotype has three attributes for specifying the priorities: *priority*, *normalizedPriority*, and *objective* (Fig. 2.20). The *priority* attribute is used to specify the priority according to the level of importance of the selection criterion. The *normalizedPriority* attribute is used to store normalized priority values which are calculated by using Equation (5). The *objective* attribute determines the target of the selection criterion. If the *objective* is given as *min*, the lowest possible value of moe is preferred; if it is given as *max*, the highest possible value of moe is preferred; if it is given a *target*, the exact value of moe is preferred.

The *EvaluationByCriteria* stereotype is applied to an element that is dedicated to be used as an executable element of a trade study in order to evaluate the alternative solution architectures by the selection criteria. The *priorityMethod* attribute is dedicated to specify the method of prioritizing the selection criteria (Linear, Pairwise, AHP) (Fig. 2.20).

Solution Architecture

A set of alternative solution architectures is provided for a trade study to evaluate them and select the most balanced solution architecture. The specified moes must be filled with a specific score based on the level of compliance with the selection criteria.

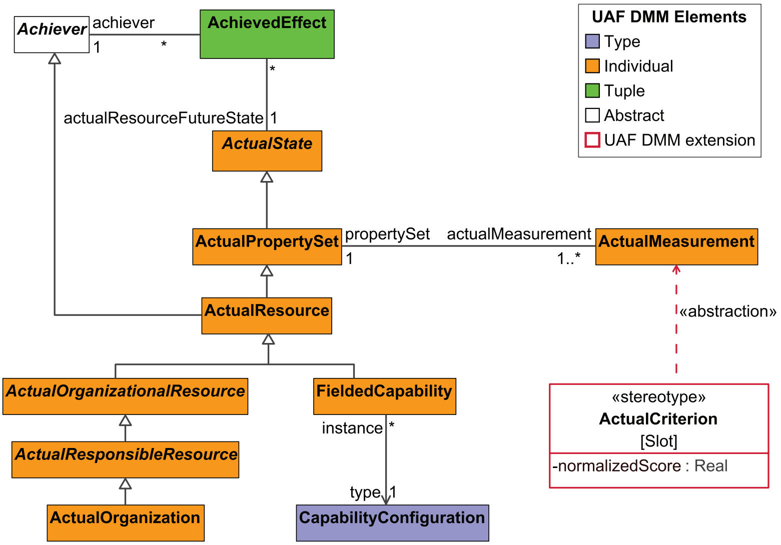


Fig. 2.21. UAF DMM scope of solution architectures

From the point of view of UAF DMM

The *FieldedCapability* instance is typed by the *CapabilityConfiguration* and is specified as the achieved configuration of the *ActualOrganization* using the *AchievedEffect* relationship. *FieldedCapability* defines a set of *ActualMeasurements* with actual values for the plausible solution architecture. The *ActualCriterion* stereotype is applied to *ActualMeasurements* in order to store a calculated normalized score of a specific selection criterion during the deep check analysis (Fig. 2.21).

2.3.2 MBSE-based architecture evaluation algorithm by selection criteria

The architecture evaluation algorithm is introduced in order to evaluate the alternative solution architectures by using the selection criteria in an automated way in the MBSE environment. In order to automatize the whole evaluation, the evaluation algorithm is based on the fUML principles.

Architecture evaluation is an iterative phenomenon in which the number of iterations depends on the number of the selection criteria (Fig. 2.22). After the selection criteria have been chosen and the weights have been determined, the scored values of each alternative solution architecture are rated. For each criterion, a rating value is assigned, and then the values are normalized.

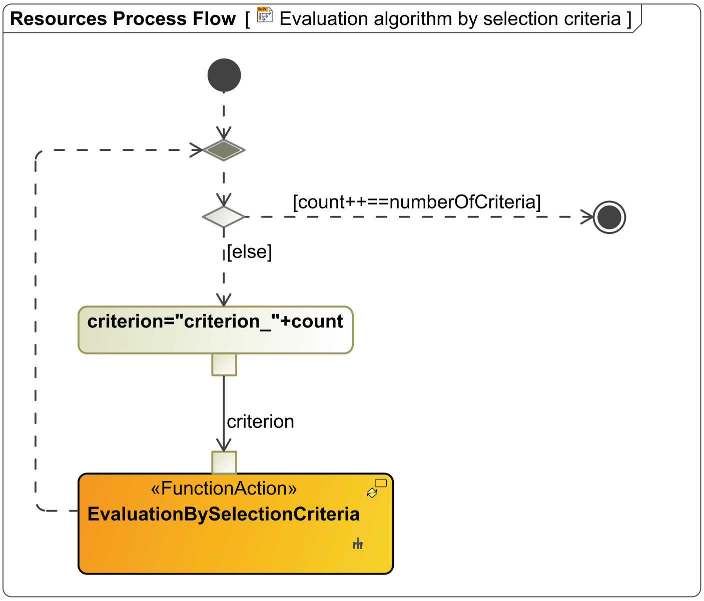

Resources Process Flow [Evaluation algorithm by selection criteria]
[count++==numberOfCriteria]
[else]
criterion="criterion_"+count
criterion
«FunctionAction»
EvaluationBySelectionCriteria

Fig. 2.22. Evaluation algorithm by selection criteria

Fig. 2.23 provides the MBSE-based architecture evolution algorithm to evaluate solution architectures against weighted selection criteria and to calculate the final score.

The evaluation starts by running two *CallBehaviorActions* (CBA) in parallel: *getScoredValuesFromAlt* and *getCriterionMetaData*. The *getScoredValuesFromAlt* CBA is called to collect the scored values of the alternative architectures according to a specific criterion. The *getCriterionMetaData* CBA is called to collect information on the currently evaluated criterion: *normalizedPriority* and *objective*.

Once the scored values of the alternative architectures and the metadata of the criterion have been obtained, the next step is to normalize the scored values. The *normalizedScoredValues* CBA is called to normalize the scored values of alternative the solution architectures in accordance with the *objective* of the criterion.

If the criterion refers to a maximum objective, normalization is calculated by using Equation (5). If the criterion refers to a minimum objective, the inverse values are calculated by using Equation (7), and then they are normalized by using Equation (5). If no objective is specified, the maximum object is used in the course of the evaluation.

$$V_{invr} = SV_{max} - (SV_{actual} - SV_{min}); \tag{7}$$

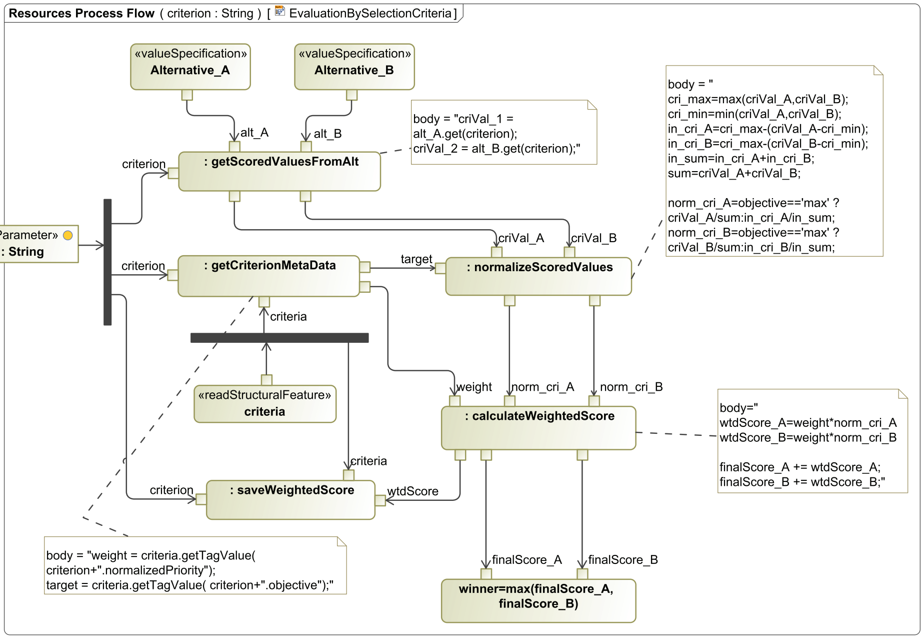
Where:

V_{invr} – Inverse value,

SV_{min} – Minimum scored value among a set of alternative solution architecture,

SV_{max} – Maximum scored value among a set of alternative solution architecture,

SV_{actual} – Actual scored value.


Resources Process Flow (criterion : String) [EvaluationBySelectionCriteria]
«valueSpecification» Alternative_A
«valueSpecification» Alternative_B
alt_A
alt_B
criterion
: getScoredValuesFromAlt
«ResourceParameter» criterion : String
criVal_A
criVal_B
criterion
: getCriterionMetaData
target
: normalizeScoredValues
criteria
«readStructuralFeature» criteria
weight
norm_cri_A
norm_cri_B
: calculateWeightedScore
criteria
criterion
: saveWeightedScore
wtdScore
finalScore_A
finalScore_B
winner=max(finalScore_A, finalScore_B)

```
body = "criVal_1 =
alt_A.get(criterion);
criVal_2 = alt_B.get(criterion);"
```

```
body = "
cri_max=max(criVal_A,criVal_B);
cri_min=min(criVal_A,criVal_B);
in_cri_A=cri_max-(criVal_A-cri_min);
in_cri_B=cri_max-(criVal_B-cri_min);
in_sum=in_cri_A+in_cri_B;
sum=criVal_A+criVal_B;

norm_cri_A=objective=='max' ?
criVal_A/sum:in_cri_A/in_sum;
norm_cri_B=objective=='max' ?
criVal_B/sum:in_cri_B/in_sum;
```

```
body="
wtdScore_A=weight*norm_cri_A
wtdScore_B=weight*norm_cri_B

finalScore_A += wtdScore_A;
finalScore_B += wtdScore_B;"
```

```
body = "weight = criteria.getTagValue(
criterion+".normalizedPriority");
target = criteria.getTagValue( criterion+".objective");"
```

Fig. 2.23. ‘EvaluationBySelectionCriteria’ Function

The next step is to calculate the weighted score based on the normalized criterion priority, and for this, the *calculateWeightedScore* CBA is called. The weighted score is calculated by using Equation (8).

$$WS = \prod_i x_i y_i; \tag{8}$$

Where:

WS – weighted score by the selection criterion,
x_i – normalized priority of ith criterion,
y_i – normalized score of ith alternative solution architecture.

Then, the *saveWeightedScore* CBA is called to save the calculated weighted score according to a specific criterion. This is necessary in order to sum up the weighted score for each alternative solution architecture by using Equation (9). Optionally, weighted scores can be converted to a scale of 100. The highest scoring alternative solution architecture has a 100-point rating, while all other alternative solution architectures are rated accordingly lower.

$$S = \sum_i^n x_i; \tag{9}$$

Where:

S – sum of the weighted scores based on all criteria,
x_i – weighted score of ith criterion.

Finally, all the weighted scores are compared, and a higher-scoring alternative solution architecture is selected as the ‘winner’ by using the *max* function.

However, when summing up the results of the architecture evaluation by the selection criteria, it is possible that the scores of several alternative solution architectures will be similar or even the same. In order to verify the choice of the preferred solution architecture, it is necessary to perform sensitivity analysis at this stage. Those alternatives that scored no worse than 5% lower than the highest-scoring alternative are selected for sensitivity analysis including the highest-scoring alternative itself [138] [139] [140] [141].

2.3.3 MBSE-based sensitivity analysis algorithm

During the sensitivity analysis, the scored values of the alternative architectures are modified to determine whether the weighted score is sensitive to the changed values. This makes it possible to identify the most sensitive criterion and then compare them with the scored values that are relevant.

Fig. 2.24 and Fig. 2.25 present the MBSE-based OAT sensitivity analysis algorithm. The OAT method evaluates the model output changes based on single-parameter input changes. The input parameters change by one, while the other parameters remain constant.

The sensitivity algorithm is similar to the MBSE-based architecture evaluation algorithm by the selection criteria. However, in the sensitivity analysis, the scored values of the alternative architectures are changed according to a specified sensitivity variable (e.g. +20%) by following the sensitivity index calculation.

The sensitivity analysis is an iterative analysis in which the number of iterations depends on the number of the selection criteria having a different actual score between other alternative solution architectures (Fig. 2.24). Only the selection criteria with different scores are included in the sensitivity analysis, as changing the criteria with the same score will lead to the same change in the weighted score.

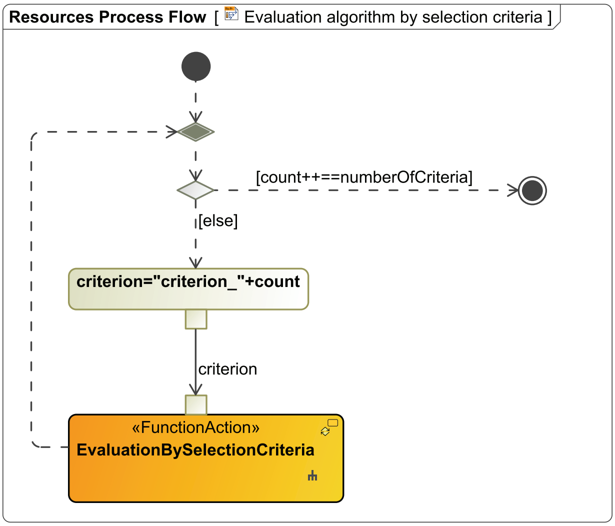
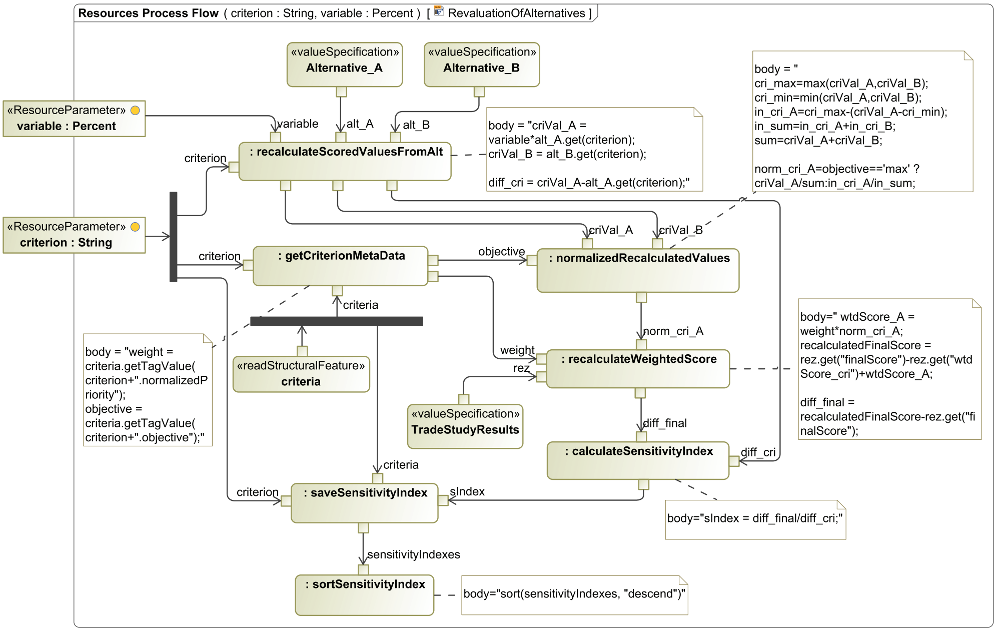

Resources Process Flow [Evaluation algorithm by selection criteria]
[count++==numberOfCriteria]
[else]
criterion="criterion_"+count
criterion
«FunctionAction»
EvaluationBySelectionCriteria

Fig. 2.24. Sensitivity analysis algorithm

Fig. 2.25 describes the MBSE-based sensitivity analysis algorithm. The evaluation starts by running two CBA in parallel: *recalculateScoredValuesFromAlt* and *getCriterionMetaData*. The *recalculateScoredValuesFromAlt* CBA is called to recalculate the scored values of the alternative architectures based on the *sensitivity variable*. The *getCriterionMetaData* CBA is called to collect information on the currently evaluated criterion.

Once the scored values of the alternative architectures have been recalculated, and the metadata of the criterion has been obtained, the next step is to normalize the recalculated scored values. The *normalizedRecalculatedValues* CBA is called to normalize the recalculated scored values by using Equations (5) or (7).


Resources Process Flow (criterion : String, variable : Percent) [RevaluationOfAlternatives]
«valueSpecification» Alternative_A
«valueSpecification» Alternative_B
«ResourceParameter» variable : Percent
«ResourceParameter» criterion : String
variable
alt_A
alt_B
criterion
: recalculateScoredValuesFromAlt
criVal_A
criVal_B
criterion
: getCriterionMetaData
objective
: normalizedRecalculatedValues
criteria
«readStructuralFeature» criteria
norm_cri_A
weight
rez
: recalculateWeightedScore
«valueSpecification» TradeStudyResults
diff_final
: calculateSensitivityIndex
diff_cri
criteria
criterion
: saveSensitivityIndex
sIndex
sensitivityIndexes
: sortSensitivityIndex

```
body = "criVal_A =
variable*alt_A.get(criterion);
criVal_B = alt_B.get(criterion);

diff_cri = criVal_A-alt_A.get(criterion);"
```

```
body = "
cri_max=max(criVal_A,criVal_B);
cri_min=min(criVal_A,criVal_B);
in_cri_A=cri_max-(criVal_A-cri_min);
in_sum=in_cri_A+in_cri_B;
sum=criVal_A+criVal_B;

norm_cri_A=objective=='max' ?
criVal_A/sum:in_cri_A/in_sum;
```

```
body = "weight =
criteria.getTagValue(
criterion+".normalizedP
riority");
objective =
criteria.getTagValue(
criterion+".objective");"
```

```
body=" wtdScore_A =
weight*norm_cri_A;
recalculatedFinalScore =
rez.get("finalScore")-rez.get("wtd
Score_cri")+wtdScore_A;

diff_final =
recalculatedFinalScore-rez.get("fi
nalScore");
```

```
body="sIndex = diff_final/diff_cri;"
```

```
body="sort(sensitivityIndexes, "descend")"
```

Fig. 2.25. ‘RevaluationOfAlternatives’ Function

The next step is to recalculate the weighted score based on the normalized criterion priority; for this, the *recalculateWeightedScore* CBA is called. The weighted score is recalculated by using Equation (8).

Afterwards, the *calculateSensitivityIndex* CBA is called to indicate the sensitivity of the weighted score to changes in the scored values by using Equation (10). The higher is the result, the greater is the sensitiveness of the criterion.

$$sensitivity_i = \frac{\Delta Y}{\Delta X_i}; \qquad (10)$$

Where:

sensitivity – sensitivity index,

Y – weighted score of the solution architecture,

X_i – i^{th} scored value of the solution architecture.

The *saveSensitivityIndex* CBA is called to save the calculated sensitivity index according to a specific criterion. In order to provide a list of sorted criteria by the sensitivity index, the *sortSensitivityIndex* CBA is called.

Finally, the responsible role should review these results to state whether the highest-scoring solution architecture identified after the evaluation by the selection criteria is in fact the most balanced solution architecture among the similar-scoring alternatives.

2.4 Comparison of UT3SA method to other trade study processes and methods

In the absence of a similar trade study method in the literature as the UT3SA method, the comparison is made fragmentally by comparing the two main parts of the UT3SA method: the trade study process and the architecture evaluation methods.

The introduced UAF-based trade study process is compared with the other trade study processes introduced by NASA, NAF, Reiter, and MITRE. The comparison criteria are as follows:

- **Defined roles** – indicates whether the process introduces specific roles in a trade study process.
- **Defined input/output data** – indicates whether the process presents the required input and output data of each step in a process.
- **Based on metamodel** – indicates whether the method is based on the metamodel.
- **Main phases of trade study process** – indicates whether the trade study process covers the main phases of the process.
    - **Preparation** – indicates whether the trade study process covers the case study preparation phase.
    - **Key criteria specification** – indicates whether the trade study process covers the specification of a set of the key criteria of a desired solution.
    - **Selection criteria specification** – indicates whether the trade study process covers the specification of the selection criteria that will be used to judge each alternative in order to select the most preferred solution.
    - **Evaluation** – indicates whether the trade study process covers the alternative evaluation steps.
    - **Step for the acquisition process** – indicates whether the trade study process includes the necessary steps when a trade study is conducted as a result of an acquisition process.
    - **Conclusion** – indicates whether the trade study process covers the trade study conclusion step by providing a formal decision and the rationale.
- **Analysis methods for evaluation phase** – descriptive (Ds) or detailed (Dt). This criterion indicates whether the evaluation methods provided by a trade study process are descriptive only, or they may include algorithms, constraints, or other methods.

- **MBSE support** – indicates whether the process provides guidance how to adapt a trade study in the MBSE environment.

Table 2.17. Comparisons of trade study processes

Criteria		NASA	NAF	Reiter	MITRE	UT3SA
Defined roles		-	-	-	-	+
Defined input/output data		-	-	-	-	+
Based on metamodel		-	+	-	-	+
Main phases of trade study process	**Preparation**	+	+	+	+	+
	Key criteria specification	-	-	-	-	+
	Selection criteria specification	+	+	+	+	+
	Evaluation	+	+	+	+	+
	Step for the acquisition process	-	-	-	-	+
	Conclusion	-	+	+	+	+
Analysis methods for the evaluation phase		Ds	Ds	Ds	Ds	Dt
MBSE support		-	-	-	-	+

The comparison results (Table 2.17) reveal that the UAF-based trade study process of UT3SA is the only process that includes roles, input/output data for each process stage, comprises the needed step for the acquisition process, provides detailed evaluation methods with algorithms and constraints, and, finally, includes the guidance on how to define the trade study in the MBSE environment. The other advantage over a few trade study processes is that the process is based on a formal metamodel.

UT3SA includes three analyses: (1) quality assessment based on constraints; (2) evaluation by selection criteria based on the AHP method; (3) sensitivity analysis based on the OAT method. The architecture evaluation methods used in UT3SA are compared with the original methods: MCDA: AHP and SA: OAT. The comparison criteria are as follows:

- **Qualitative and quantitative analysis of SoS architecture parameters** – indicates whether a method can be used to evaluate the qualitative and quantitative parameters of a system.
- **Correctness & completeness analysis of SoS architecture model** – indicates whether a method can be used to assess the correctness and completeness of the SoS architecture model.
- **Quantitative measurement** – indicates whether a method produces a quantitative measurement result that can be represented numerically.
- **MBSE support** – indicates whether a method includes the needed guidance on how to adopt it in the MBSE environment.

Table 2.18. Comparison of architecture evaluation methods

Criteria	**MCDA: AHP + SA: OAT**	**UT3SA**
Qualitative and quantitative analysis of SoS architecture parameters	+	+
Correctness & completeness analysis of SoS architecture model	-	+
Quantitative measurement	+	+
MBSE support	-	+ (Algorithms are included)

The comparison results (Table 2.18) revealed that the UT3SA architecture evaluation methods are superior to the combination of the AHP and OAT methods. UT3SA evaluation methods provide two unique features: (1) support of correctness and completeness analysis of the SoS architecture model; (2) MBSE support.

2.5 Summary of the UT3SA method

Analysis of the already existing ADFs, trade study processes and evaluation methods has revealed gaps between them. Furthermore, it has exposed the complexity of the process and evaluation method application, especially for SoS architectures in the MBSE environment. To fill this gap, a trade study method (called UT3SA) has been introduced that is strictly based on the UAFP principles.

The UT3SA method consists of two main parts: (1) UAF-based trade study process; (2) UAF-based architecture evaluation algorithms and guidelines in the MBSE environment. The UAF-based trade study process defines the application of the UT3SA method, including the roles and the input/output data for each stage. While the second part of the UT3SA method provides the necessary algorithms for the evaluation of UAF-based architectures by running three analyses: (1) quality assessment, the output of the analysis is the quality index; (2) evaluation by selection criteria, the output of the analysis is the weighted score; (3) sensitivity analysis, the output of the analysis is the sensitivity index.

As the UT3SA method is analytical, it is important to ensure that the weighted selection criteria and the alternative solution architectures are appropriately specified in the MBSE environment before applying this method. For this reason, the UT3SA method guides how a certain part of the architecture evaluation should be specified. Additionally, the missing trade study concepts are introduced as extensions of UAF DMM that are needed to automate the trade study in the MBSE environment.

In the absence of a similar trade study method in the literature to the UT3SA method, a comparison of UT3SA was fragmentally made by comparing two main parts: the process and architecture evaluation methods. The comparison revealed that the UAF-based trade study process of UT3SA is the only process with five unique features: (1) defines roles; (2) defines input/output data for each process stage; (3) comprises the step needed for the acquisition process; (4) provides detailed evaluation methods with algorithms and constraints; (5) includes the guidance on how to define the trade study in the MBSE environment. After the comparison of the architecture evaluation methods, it was indicated that the UT3SA architecture evaluation methods are superior to other, alternative, existing methods by providing two unique features: (1) support of correctness and completeness analysis of the SoS architecture model; (2) MBSE support.

The following chapter presents an evaluation of the suitability of the proposed UT3SA method while applying it to the trade study of eight alternative solution architectures.

3 EXPERIMENTAL EVALUATION OF UT3SA METHOD

This chapter includes three parts. The first part describes the implementation of the UT3SA method by introducing the UAF profile, a set of constraints and validation-based metrics, a trade study pattern and analysis tables. The second part presents an experiment applying the UT3SA method to a requirements management (RM) tool trade study with eight alternative RM tools. The third part presents the conclusions of the experimental evaluation of the UT3SA method.

Only one experimental evaluation is detailed in this dissertation. However, the other experimental evaluations of the proposed UT3SA method are provided in the published journal and conference proceedings. The UT3SA method was applied in the following areas: electric road configuration [131], and marine search and rescue study [134], [56].

3.1 Experimental environment for evaluating UT3SA

This section describes the UT3SA implementation. UT3SA is defined in the form of a UAF plug-in for the UAF Case tool, including a UAF profile and a set of constraints.

3.1.1 UT3SA plug-in for UAF Case tool

Fig. 3.1 provides the necessary deployment configuration for the experimental evaluation of the UT3SA method. The general configuration describes a typical environment configuration consisting of a personal computer with the UAF Case tool installed. The UAF Case tool must support fUML and executable OCL constraints. The UT3SA plug-in contains the UAF profile for the UT3SA method, including constraints.

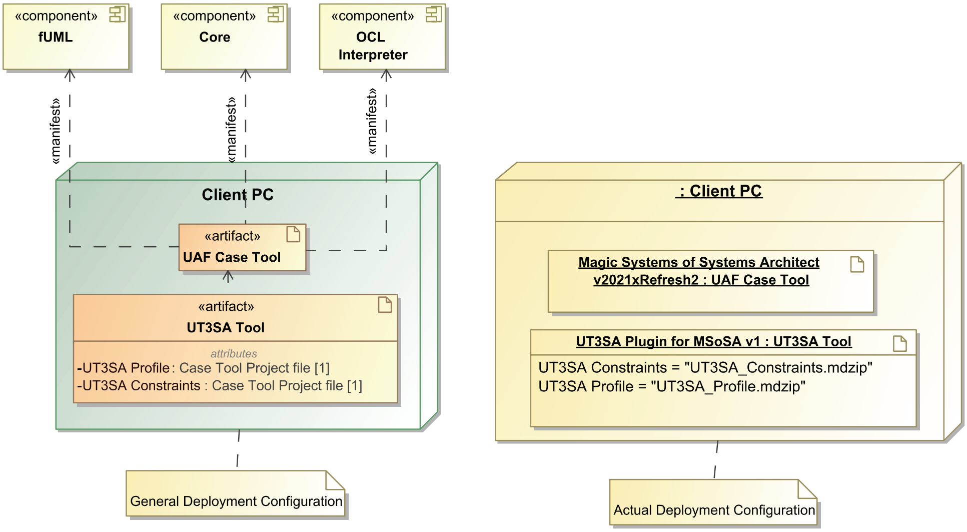

«component»
fUML
«component»
Core
«component»
OCL Interpreter
«manifest»
«manifest»
«manifest»
Client PC
«artifact»
UAF Case Tool
«artifact»
UT3SA Tool
attributes
-UT3SA Profile : Case Tool Project file [1]
-UT3SA Constraints : Case Tool Project file [1]
: Client PC
Magic Systems of Systems Architect v2021xRefresh2 : UAF Case Tool
UT3SA Plugin for MSoSA v1 : UT3SA Tool
UT3SA Constraints = "UT3SA_Constraints.mdzip"
UT3SA Profile = "UT3SA_Profile.mdzip"
General Deployment Configuration
Actual Deployment Configuration

Fig. 3.1. Deployment configuration of UT3SA method

The actual deployment configuration indicates the actual environment configuration used to carry out experimental evaluation of the UT3SA method. The experiment is carried out by using the *Catia Magic Systems of Systems Architect* v2021xRefresh2 Case tool and the *UT3SA plug-in for MSoSA* v1.00.

3.1.2 UT3SA profile

In order to implement the proposed method in one of the UAF CASE tools, first, a UAF profile should be developed. Second, additional stereotypes are required to define the selection criteria for a trade study and perform analysis by using simulation techniques. Table 3.1 provides the necessary stereotypes for a trade study which are contained in the UT3SA profile.

Table 3.1. Description of the stereotypes in UT3SA profile

Stereotype	UML Metaclass	Attributes	Description
Critical	Class Activity Property	-	The *Critical* stereotype refers to capabilities, operational activities and measurements that are identified as critical to the trade study. Each alternative solution architecture must meet all the critical requirements; otherwise, it is excluded from the trade study.
PrioritizedRequirement	Class	*priority* *normalizedPriority* *objective*	The *PrioritizedRequirement* stereotype refers to requirements dedicated to the trade study. This stereotype has three attributes: *priority*, *normalizedPriority*, and *objective*. The *priority* attribute is dedicated to storing the priority level of a specific requirement, which is considered a selection criterion or a group of selection criteria. The *normalizedPriority* attribute is dedicated to storing the normalized priority value of a specific selection criterion group. This attribute should only be used when a trade study is carried out with groups of selection criteria.

Stereotype	UML Metaclass	Attributes	Description
			The *objective* attribute is dedicated to storing the target of the selection criterion: min, max, target.
Criterion	Property	*/weight*	The *Criterion* stereotype refers to measurements dedicated to the trade study. This stereotype has only one attribute: *weight*. The *Criterion* stereotype extends a property metaclass to store required data in a model so that to evaluate alternative architectures by using simulation. The *weight* is a derived attribute that is intended to store the level of importance of the trade study selection criterion, which corresponds to the priority value of the *PrioritizedRequirement.*
ActualCriterion	Slot	*normalizedScore*	The *ActualCriterion* stereotype refers to actual measurements of an alternative solution architecture. This stereotype has one attribute: *normalizedWeight*. The *ActualCriterion* stereotype extends a slot metaclass to store required data in a model in order to evaluate alternative architectures using simulation. The *normalizedScore* attribute is dedicated to storing the normalized score of a specific selection criterion.
EvaluationByCriteria	Class	*results* *winner* *score* *alternativesForSensitivityAnalysis* *priorityMethod*	The *EvaluationByCriteria* stereotype is introduced to indicate an element that is dedicated to be used as an executable element of trade study and to evaluate alternative solution architectures by

Stereotype	UML Metaclass	Attributes	Description
			selection criteria. This stereotype has five attributes: *results*, *winner*, *score, alternativesForSensitivityAnalysis* and priorityMethod. The *results* attribute is dedicated to store a set of final scores of all alternative solution architectures evaluated. The *winner* attribute is dedicated to store one alternative solution architecture that gets the highest final score. The *score* attribute is dedicated to store the final score of the alternative solution architecture presented at the 'winner' attribute. The *alternativesForSensitivityAnalysis* attribute is dedicated to store the set of alternative solution architectures that have the same or similar final score as the 'winner' alternative. The *priorityMethod* attribute is dedicated to store the selected method to specify priority level for selection criteria (linear, pairwise, AHP).
SensitivityAnalysis	Class	*sensitivityVariable* *alternatives*	The *SensitivityAnalysis* stereotype is introduced to indicate an element that is dedicated to be used as an executable element of trade study and to perform sensitivity analysis. This stereotype has two attributes: s*ensitivityVariable* and a*lternatives.* The s*ensitivityVariable* attribute is dedicated to store the

Stereotype	UML Metaclass	Attributes	Description
			percentage used to increase/decrease the actual score of the alternative solution architecture. The a*lternatives* attribute is dedicated to provide the list of alternatives that will be analyzed during the sensitivity analysis.

UT3SA stereotypes can be applied to any element that has a corresponding metaclass, so it can be used by employing any modeling method and language supported by the CASE tool (e.g. BPMN, TOGAF, UPDM, etc.).

Additionally, in order to perform the trade study analysis and to review its results in a more convenient way, two tables with custom columns are introduced. Table 3.2 describes two analysis tables for a trade study which are contained in the UT3SA profile.

Table 3.2. Description of the analysis tables in UT3SA profile

Table	Custom Columns	Description
EvaluationByCriteriaTable	*CapabilityConfiguration* *SelectionCriteria* *Alternatives*	The *EvaluationByCriteriaTable* table is introduced to perform evaluation of alternative solution architectures by the selection criteria. This table has three custom columns: *CapabilityConfiguration*, *SelectionCriteria* and *Alternatives*. The *CapabilityConfiguration* custom column is dedicated to present a *CapabilityConfiguration* element that has determined selection criteria for the trade study. The *SelectionCriteria* custom column is dedicated to present a list of selection criteria based on *CapabilityConfiguration*. The *Alternatives* custom column is dedicated to present a list of plausible alternative solution architectures.

Table	Custom Columns	Description
SensitivityAnalysisTable	*CapabilityConfiguration* *AnalysisCriteria*	The *SensitivityAnalysisTable* table is introduced to perform sensitivity analysis for those alternative solution architectures whose final score is similar to the ‘winner’ alternative. This table has two custom columns: *CapabilityConfiguration* and *AnalysisCriteria*. The *CapabilityConfiguration* custom column is dedicated to present a *CapabilityConfiguration* element which possesses determined selection criteria for the trade study. The *AnalysisCriteria* custom column is dedicated to present a list of sensitivity analysis selection criteria. This custom column lists the selection criteria whose actual score is different between the alternatives being analyzed.

The table is one of the possible options for reviewing analysis results in the MSoSA tool. In addition, the results can be reviewed from an executable element with the *EvaluationByCriteria* or *SensitivityAnalysis* stereotype applied, or with a report generated.

3.1.3 UT3SA constraints

In order to perform the quality assessment at the base check stage, the validation rules are introduced in the UT3SA plug-in based on the description provided in Section 2.2.2. Two different severity levels are used for the rules: warning and error. The warning severity level is used for the validation rules whose weight is equal to one. The error severity level is used for the validation rules whose weight is equal to two or three. The weight of the rule is a size that can be freely selected. The higher is the weight, the more important the rule is compared to other rules. The validation rules are stored in the validation suite of ‘UT3SA_Constraints.mdzip’. The validation suite in the MSoSA tool means that the rules are run by manually invoking them.

To perform the quality assessment and review its results in a more convenient way, three metric suites are introduced in the ‘UT3SA_Constraints.mdzip’ (Fig. 3.2).

The metric suites (‘Op vs. Rs’, ‘Horizontal’, ‘St vs. Rs’) are created according to the defined structure of the validation rules set in (Fig. 2.10).

Each metric suite contains validation-based metric definitions that are linked to a particular validation rule. For one validation rule, three validation-based metric definitions are defined: (1) count the quantity of evaluated elements; (2) count the quantity of the failed elements; (3) count the percentage of the failed elements. Each metric suite has a defined metric definition ‘partialQualityIndex’ that is dedicated to calculate the quality index within the scope of a particular metric suite by using Equation (4).

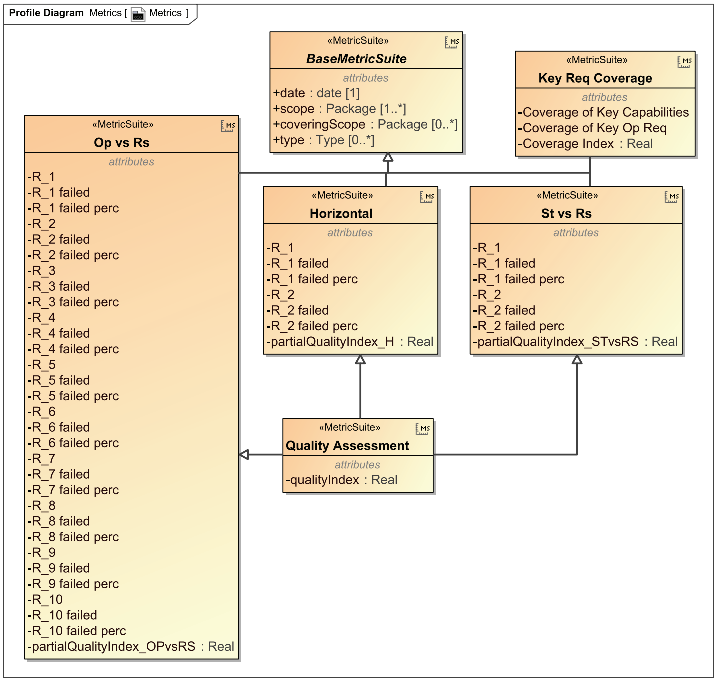


Fig. 3.2. Validation-based metric definitions for UT3SA method

In order to see all the quality assessment results of the evaluated alternatives in one place, a special metric suite called ‘Quality Assessment’ is introduced. The Quality Assessment has a defined metric definition ‘qualityIndex’ that is dedicated to calculate the final quality index of quality assessment according to the separately calculated partial quality indexes.

Additionally, the ‘Key Req Coverage’ metric suite is introduced to check the key requirements coverage at the base check stage. The ‘Key Req Coverage’ metric suite contains two validation-based metric definitions that are dedicated to calculate the key capabilities and the operational requirements coverages separately. The ‘Coverage Index’ metric definition calculates the total coverage of the key capabilities and operational requirements.

To review the results of the calculated metrics based on the validation rules, the metric table should be used in the MSoSA tool. In addition, the results can be seen through a generated report.

3.2 UT3SA method evaluation

In order to confirm the suitability of the proposed method, UT3SA is used in a trade study of the RM tool. The trade study of the RM tools is chosen because more and more organizations are adopting requirements tools as they are seeking support in managing requirements information, in traceability to ensure that the scope is controlled, and in modeling with the objective to visually represent the requirements. The purpose of this trade study is to choose the preferred RM tool for an existing Case tool based on the defined set of selection criteria.

The aim of the experiment is to perform a trade study of the solution architectures by using the UT3SA method. The experimental environment is based on the actual deployment configuration defined in Section 3.1.1.

Additionally, seven human experts were included in the experiment. The experts were selected for the experiment according to the following criteria: (1) knowledge of RM procedures and/or RM tools; (2) 5+ years of experience in MBSE/SE/SoS; (3) experience in developing complex systems; (4) UML/SysML/BPMN certificate. Adherence to these criteria ensures the sufficiency of expert knowledge to conduct the RM tool trade study and participate in the reliability and validity check of the experiment results.

3.2.1 Process for evaluating UT3SA method

Fig. 3.3 shows the evaluation process of the UT3SA method. The evaluation process is organized according to the UT3SA process (Fig. 2.2), but focuses mainly on the steps that are dedicated to the evaluation of the alternative solution architectures.

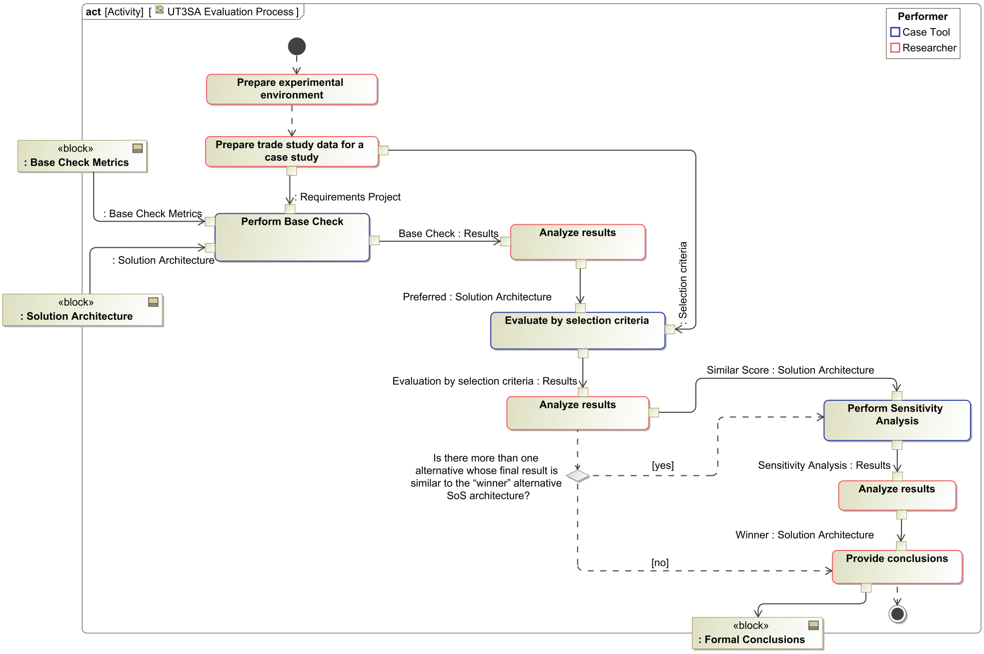

act [Activity] [UT3SA Evaluation Process]
Performer
□ Case Tool
□ Researcher
Prepare experimental environment
Prepare trade study data for a case study
«block»
: Base Check Metrics
: Requirements Project
: Base Check Metrics
Perform Base Check
: Solution Architecture
«block»
: Solution Architecture
Base Check : Results
Analyze results
Preferred : Solution Architecture
Evaluate by selection criteria
: Selection criteria
Evaluation by selection criteria : Results
Analyze results
Similar Score : Solution Architecture
Perform Sensitivity Analysis
Is there more than one alternative whose final result is similar to the "winner" alternative SoS architecture?
[yes]
Sensitivity Analysis : Results
Analyze results
Winner : Solution Architecture
[no]
Provide conclusions
«block»
: Formal Conclusions

Fig. 3.3. Evaluation process of UT3SA method

Each step of the evaluation process of the UT3SA method is detailed in Table 3.3.

Table 3.3. Definitions of tasks of UT3SA evaluation process

Name	**Property**	**Description**
Prepare experimental environment	Definition	It is a manual task for the human resource to prepare the experimental environment for the evaluation of the UT3SA method
	Preconditions	– MSoSA v2021xRefresh2 installed. – Magic Model Analyst v2021xRefresh2 installed. – UT3SA Plug-in v1.00 installed
	Performer	Researcher
Prepare trade study data for a case study	Definition	It is a manual task for the human resource to define the logical architecture (Op domain), identify the key requirements (St domain) of a preferred system and the selection criteria with the assigned priority level
	Outputs	Project with defined Op and St domains and established selection criteria
	Performer	Researcher
Perform Base Check	Definition	The Base Check of the UT3SA is performed, which is defined in Section 2.2
	Inputs	Project with defined Op and St domains (requirements project). List of selected alternative solution architectures. Set of Base Check validation-based metrics
	Outputs	Detailed results of performed Base Check analysis
	Performer	Case tool
Analyze results	Definition	It is manual task for the human resource to analyze the results of the Base Check and define a set of plausible alternative SoS solution architectures
	Inputs	Detailed results of performed Base Check analysis
	Outputs	List of plausible alternative solution architectures
	Performer	Researcher
Evaluate by selection criteria	Definition	The evaluation by selection criteria of the UT3SA is performed, which is defined in Section 2.3.2
	Inputs	List of plausible alternative solution architectures. Set of trade study selection criteria
	Outputs	Detailed results of performed evaluation by selection criteria
	Performer	Case tool
Analyze results	Definition	It is manual task for the human resource to analyze the results of the evaluation by selection criteria and choose the alternatives that have received a similar or equal final score.
	Inputs	Detailed results of the performed evaluation by the selection criteria

Name	Property	Description
	Outputs	List of alternatives with similar final score
	Performer	Researcher
Perform Sensitivity Analysis	Definition	Sensitivity Analysis of UT3SA is performed, which is defined in Section 2.3.3
	Inputs	List of alternatives with a similar final score
	Outputs	Detailed results of the performed sensitivity analysis
	Preconditions	There is more than one alternative with a similar score to the 'winner' alternative
	Performer	Case tool
Analyze results	Definition	It is a manual task for the human resource to analyze the results of the sensitivity analysis and choose the 'winner' solution architecture
	Inputs	Detailed results of the performed sensitivity analysis
	Outputs	The 'winner' alternative RM solution architecture
	Performer	Researcher
Provide conclusions	Definition	It is a manual task for the human resource to provide the key decision and rationale as to which alternative solution architecture is the most balanced one for the preferred system
	Inputs	The 'winner' alternative RM solution architecture. Detailed results of the performed sensitivity analysis
	Outputs	Formal conclusions
	Performer	Researcher

In the following section, the case study for applying the UT3SA method is described.

3.2.2 Case Study

In this section, the case study for the proposed UT3SA method is presented. The trade study of the requirements management tool is selected as the target of the case study. The trade study is performed by evaluating eight alternative solution architectures of different RM tools in order to select the preferred solution architecture. This case study includes five main steps: (1) data preparation of the trade study including the selection criteria and the alternative solution architecture; (2) base check analysis; (3) evaluation of alternative solution architectures by the selection criteria; (4) sensitivity analysis; (5) review of the results and the final decision.

The aim of the case study is to provide an example of UT3SA application for the SoS architecture in the MBSE environment.

Preparation of trade study data

A traditional SE company is chosen as the subject of this trade study, which is further called 'Enterprise X'. Fig. 3.4 shows the main structure of Enterprise X. As the trade study is performed in order to find the preferred RM tool among the available alternatives, the ‘Requirements Management’ *Capability* of ‘R&D’ *Capability* is mainly focused on this trade study.

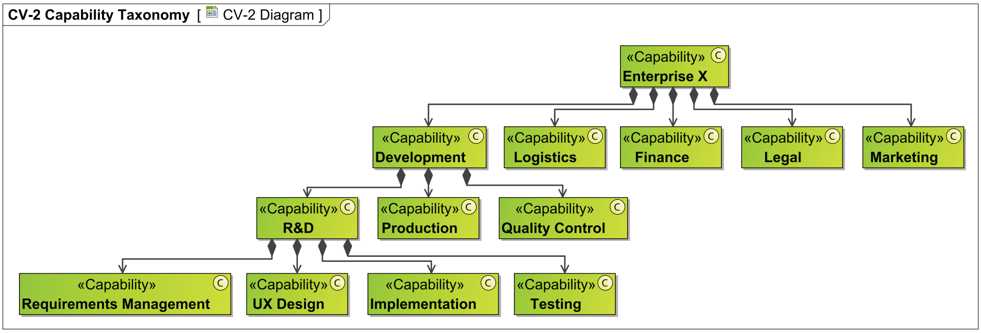


Fig. 3.4. Capability taxonomy diagram of Enterprise X

Currently, Enterprise X is using MS Excel as a tool for requirements management. However, due to the manual requirement tracking process, many versions of the same requirements specification document, incomplete, incorrect, and lost requirements, it has been decided to change the current RM tool into a more sophisticated tool.

Fig. 3.5 shows two phases of Enterprise X, the ‘As Is’ phase provided on the left side, and the ‘To Be’ phase provided on the right side, which shows the required capabilities of the preferred RM tool. The defined ‘Requirement Management – Sophisticated Tool’ *ActualEnterprisePhase* should exhibit nine capabilities: Planning, Elicitation, Analysis, Specification, Validation, Management, Tool Administration, Usability, Integrations. The newly introduced capabilities are marked with the red border, whereas the capabilities that are more sophisticated if compared to the ‘Requirement Management – Excel’ phase are marked with the blue border.

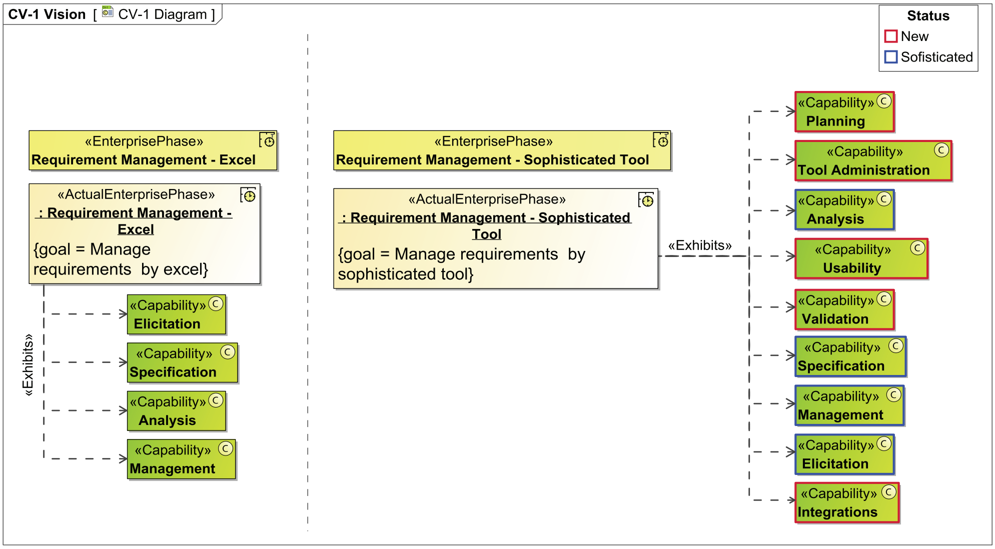
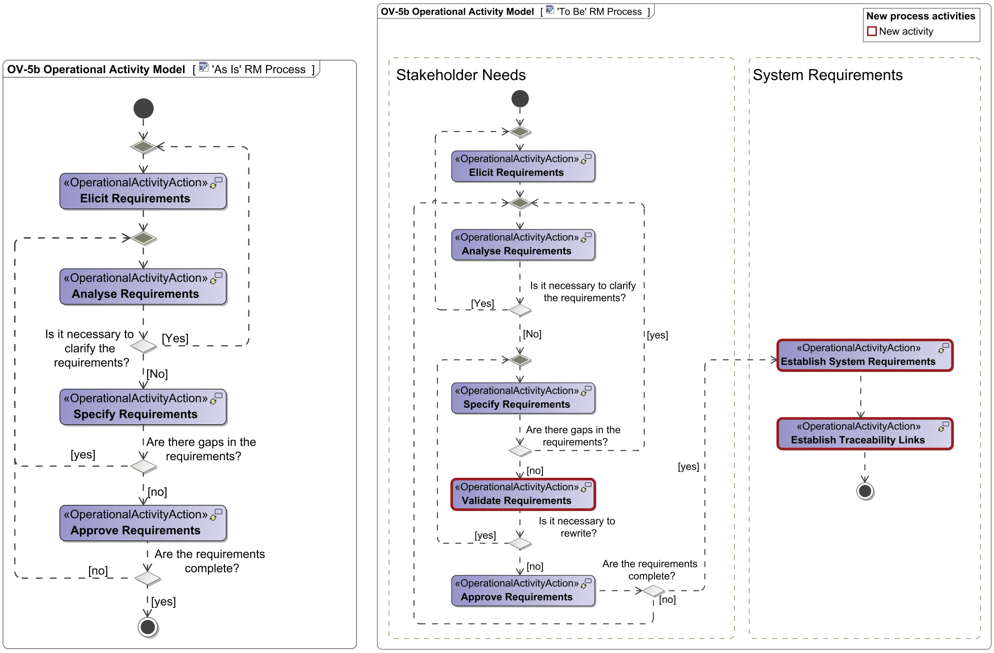

CV-1 Vision [CV-1 Diagram]
Status
□ New
□ Sofisticated
«EnterprisePhase» Requirement Management - Excel
«ActualEnterprisePhase» : Requirement Management - Excel
{goal = Manage requirements by excel}
«Exhibits»
«Capability» Elicitation
«Capability» Specification
«Capability» Analysis
«Capability» Management
«EnterprisePhase» Requirement Management - Sophisticated Tool
«ActualEnterprisePhase» : Requirement Management - Sophisticated Tool
{goal = Manage requirements by sophisticated tool}
«Exhibits»
«Capability» Planning
«Capability» Tool Administration
«Capability» Analysis
«Capability» Usability
«Capability» Validation
«Capability» Specification
«Capability» Management
«Capability» Elicitation
«Capability» Integrations

Fig. 3.5. Vision diagram of Enterprise X

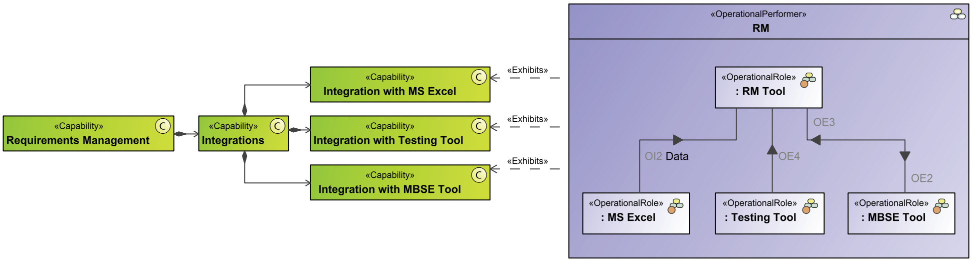
Fig. 3.6 presents two requirements management processes of Enterprise X, the 'As Is' RM process provided on the left side, and the 'To Be' RM process on the right side. The 'As Is' RM process does not distinguish the stakeholder needs from the system requirements due to the impossibility of establishing traceability relationships. The 'To Be' RM process introduces a clear distinction between the establishment of the stakeholder needs and the system requirements in the main RM process. Three new activities are introduced in the 'To Be' RM process: validate requirements, establish system requirements, and establish traceability links.



OV-5b Operational Activity Model ['As Is' RM Process]
«OperationalActivityAction» Elicit Requirements
«OperationalActivityAction» Analyse Requirements
Is it necessary to clarify the requirements?
[Yes]
[No]
«OperationalActivityAction» Specify Requirements
Are there gaps in the requirements?
[yes]
[no]
«OperationalActivityAction» Approve Requirements
Are the requirements complete?
[no]
[yes]
OV-5b Operational Activity Model ['To Be' RM Process]
New process activities
New activity
Stakeholder Needs
System Requirements
«OperationalActivityAction» Elicit Requirements
«OperationalActivityAction» Analyse Requirements
Is it necessary to clarify the requirements?
[Yes]
[No]
[yes]
«OperationalActivityAction» Specify Requirements
Are there gaps in the requirements?
[no]
[yes]
«OperationalActivityAction» Validate Requirements
Is it necessary to rewrite?
[yes]
[no]
«OperationalActivityAction» Approve Requirements
Are the requirements complete?
[no]
«OperationalActivityAction» Establish System Requirements
«OperationalActivityAction» Establish Traceability Links

Fig. 3.6. 'As Is' and 'To Be' RM Processes

Enterprise X uses various specific tools during the development process, such as the MBSE tool, the testing tool, and MS Excel. Thus, the preferred RM tool should not only introduce new activities into the RM process, but should also have the interfaces to communicate with the other tools included in the development process. Fig. 3.7 shows the necessary interfaces of the preferred RM tool, as well as indicates the transmitted data and its direction.



«Capability» Requirements Management
«Capability» Integrations
«Capability» Integration with MS Excel
«Capability» Integration with Testing Tool
«Capability» Integration with MBSE Tool
«Exhibits»
«Exhibits»
«Exhibits»
«OperationalPerformer» RM
«OperationalRole» : RM Tool
OE3
OI2 Data
OE4
OE2
«OperationalRole» : MS Excel
«OperationalRole» : Testing Tool
«OperationalRole» : MBSE Tool

Fig. 3.7. Resources internal connectivity diagram of RM Tool Configuration

Once the logical architecture has been defined, the next step is to identify the key capabilities, operational requirements and measurements for the trade study that must be met by the preferred RM tool.

Identify critical capabilities and operational requirements

If there are capabilities and operational requirements that are essential for the preferred functionality, they have to be identified. For that, the *Critical* stereotype is used. Fig. 3.8 shows three critical capabilities of the Integrations group: (1) Integration with the MBSE Tool; (2) Integration with MS Excel; (3) Integration with the Testing Tool. The 'Establish Traceability Links' *OperationalActivity* is identified as a critical operational requirement for this trade study.

In order to include critical quantitative criteria, e.g. the cost of the RM tool, the 'RM Tool Configuration' *Capability Configuration* is created, which is associated with the 'Requirements Management' *Capability* by the *Exhibits* relationship. The 'cost' measurement with the applied *Critical* stereotype is defined for the 'RM Tool Configuration' *Capability Configuration*. In addition, the 'RM Tool Cost' requirement is established, which satisfies the 'cost' measurement and indicates the limits.

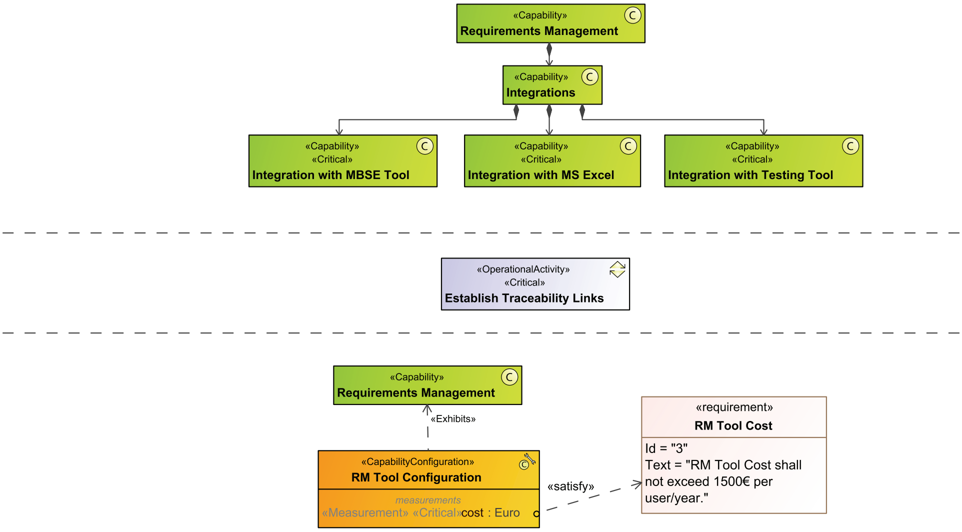


Fig. 3.8. Critical capabilities and operational requirements

Capabilities, *OperationalActivities* and *Measurements* that have applied the *Critical* stereotype must be satisfied; otherwise, the alternative RM solution architecture will be immediately excluded from the trade study.

Define Selection Criteria

To define the selection criteria of a trade study, the *requirement* element is used. The *PrioritizedRequirement* stereotype is applied to the requirements groups to the indicated requirements for the trade study and to store the calculated weighting factor of a particular selection criteria group.

Fig. 3.9 shows that each of the nine defined capabilities is refined by a requirement which represents the group of requirements of the scope of a particular capability. As there are many capabilities, the usage of the requirements groups allows to have a more organized trade study model, and, on top of that, the requirements groups can be prioritized. In addition, the use of prioritized requirement groups in the trade study provides a more accurate final score for each evaluated alternative solution architecture, as the final score calculation includes the priorities of the requirement group.

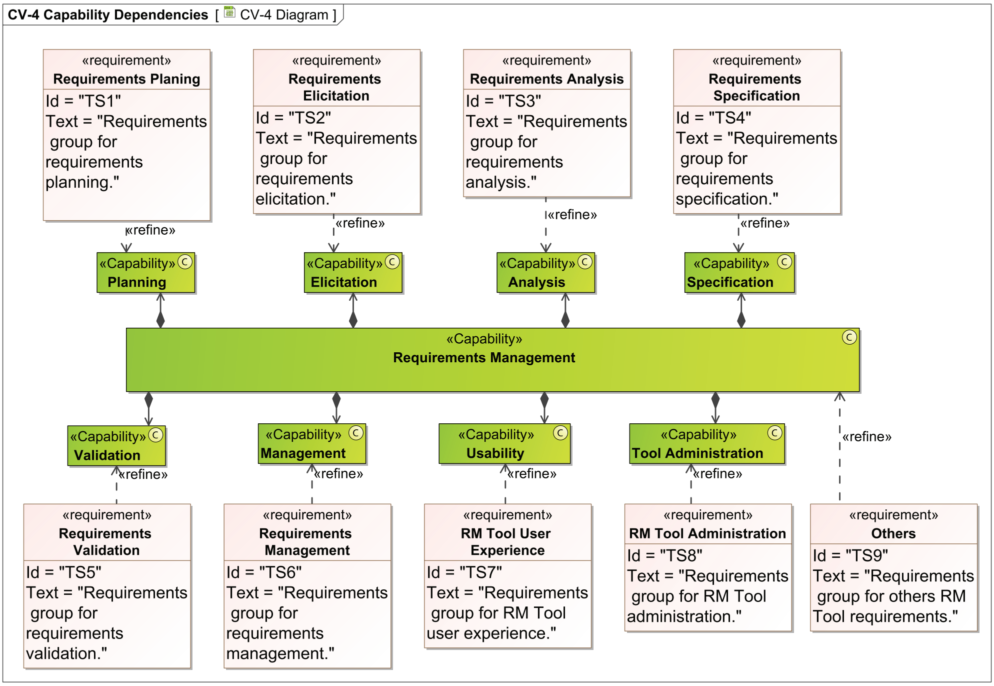

CV-4 Capability Dependencies [CV-4 Diagram]
«requirement» Requirements Planing
Id = "TS1"
Text = "Requirements group for requirements planning."
«requirement» Requirements Elicitation
Id = "TS2"
Text = "Requirements group for requirements elicitation."
«requirement» Requirements Analysis
Id = "TS3"
Text = "Requirements group for requirements analysis."
«requirement» Requirements Specification
Id = "TS4"
Text = "Requirements group for requirements specification."
«refine»
«Capability» Planning
«Capability» Elicitation
«Capability» Analysis
«Capability» Specification
«Capability» Requirements Management
«Capability» Validation
«Capability» Management
«Capability» Usability
«Capability» Tool Administration
«refine»
«requirement» Requirements Validation
Id = "TS5"
Text = "Requirements group for requirements validation."
«requirement» Requirements Management
Id = "TS6"
Text = "Requirements group for requirements management."
«requirement» RM Tool User Experience
Id = "TS7"
Text = "Requirements group for RM Tool user experience."
«requirement» RM Tool Administration
Id = "TS8"
Text = "Requirements group for RM Tool administration."
«requirement» Others
Id = "TS9"
Text = "Requirements group for others RM Tool requirements."

Fig. 3.9. Capabilities and requirements group dependencies

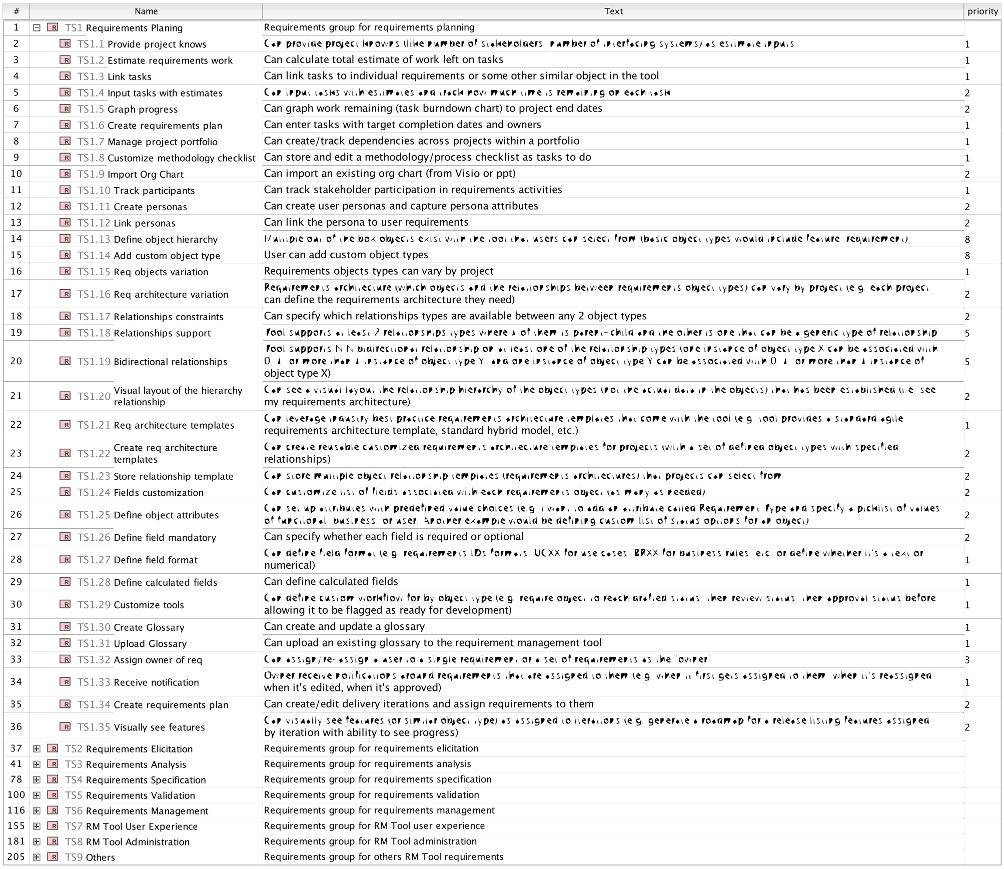
The next step is to establish a set of requirements with the required functionality and characteristics for the defined requirements groups that the preferred RM tool should meet. For that, the *requirements* elements are created and linked to the specific requirements group element by using the containment relationship. In addition, the *PrioritizedRequirement* stereotype is applied for the requirements to the indicated requirements that are dedicated for a trade study. Then, the priority level of a certain requirement is determined by using the *priority* tag of the *PrioritizedRequirement* stereotype. In this case study, the priority level is set according to Table 3.4. The priority level scale can be freely selected according to the researcher.

Table 3.4. Priority levels of requirements

Priority	Description
8	Important high-level feature
5	Important medium-level feature
3	Important low-level feature

Priority	Description
2	Performance feature, which is important in increasing the effectiveness in requirements management
1	Nice to have but not critical for an RM tool

In total, 207 selection criteria are defined (Requirements Planning – 35; Requirements Elicitation – 3; Requirements Analysis – 36; Requirements Specification – 21; Requirements Validation – 15; Requirements Management – 38; RM Tool Usability – 25; RM Tool Administration – 23; Others – 11). The complete list of the defined selection criteria for a trade study of the RM tool is provided in Appendix A.



#	Name	Text	priority
1	⊟ R TS1 Requirements Planing	Requirements group for requirements planning	
2	R TS1.1 Provide project knows	Can provide project knowns (like number of stakeholders, number of interfacing systems) as estimate inputs	1
3	R TS1.2 Estimate requirements work	Can calculate total estimate of work left on tasks	1
4	R TS1.3 Link tasks	Can link tasks to individual requirements or some other similar object in the tool	1
5	R TS1.4 Input tasks with estimates	Can input tasks with estimates and track how much time is remaining on each task	2
6	R TS1.5 Graph progress	Can graph work remaining (task burndown chart) to project end dates	2
7	R TS1.6 Create requirements plan	Can enter tasks with target completion dates and owners	1
8	R TS1.7 Manage project portfolio	Can create/track dependencies across projects within a portfolio	1
9	R TS1.8 Customize methodology checklist	Can store and edit a methodology/process checklist as tasks to do	1
10	R TS1.9 Import Org Chart	Can import an existing org chart (from Visio or ppt)	2
11	R TS1.10 Track participants	Can track stakeholder participation in requirements activities	1
12	R TS1.11 Create personas	Can create user personas and capture persona attributes	2
13	R TS1.12 Link personas	Can link the persona to user requirements	2
14	R TS1.13 Define object hierarchy	Multiple out of the box objects exist with the tool that users can select from (basic object types would include feature, requirement)	8
15	R TS1.14 Add custom object type	User can add custom object types	8
16	R TS1.15 Req objects variation	Requirements objects types can vary by project	1
17	R TS1.16 Req architecture variation	Requirements architecture (which objects and the relationships between requirements object types) can vary by project (e.g. each project can define the requirements architecture they need)	2
18	R TS1.17 Relationships constraints	Can specify which relationships types are available between any 2 object types	2
19	R TS1.18 Relationships support	Tool supports at least 2 relationships types where 1 of them is parent-child and the other is one that can be a generic type of relationship	5
20	R TS1.19 Bidirectional relationships	Tool supports N:N bidirectional relationship on at least one of the relationship types (one instance of object type X can be associated with 0, 1, or more than 1 instance of object type Y, and one instance of object type Y can be associated with 0, 1, or more than 1 instance of object type X)	5
21	R TS1.20 Visual layout of the hierarchy relationship	Can see a visual layout the relationship hierarchy of the object types (not the actual data in the objects) that has been established (i.e. see my requirements architecture)	2
22	R TS1.21 Req architecture templates	Can leverage industry best practice requirements architecture templates that come with the tool (e.g. tool provides a standard agile requirements architecture template, standard hybrid model, etc.)	1
23	R TS1.22 Create req architecture templates	Can create reusable customized requirements architecture templates for projects (with a set of defined object types with specified relationships)	2
24	R TS1.23 Store relationship template	Can store multiple object relationship templates (requirements architectures) that projects can select from	2
25	R TS1.24 Fields customization	Can customize list of fields associated with each requirements object (as many as needed)	2
26	R TS1.25 Define object attributes	Can set up attributes with predefined value choices (e.g. I want to add an attribute called Requirement Type and specify a picklist of values of functional, business, or user. Another example would be defining custom list of status options for an object)	2
27	R TS1.26 Define field mandatory	Can specify whether each field is required or optional	2
28	R TS1.27 Define field format	Can define field format (e.g. requirements IDs formats, UCXX for use cases, BRXX for business rules, etc. or define whether it's a text or numerical)	1
29	R TS1.28 Define calculated fields	Can define calculated fields	1
30	R TS1.29 Customize tools	Can define custom workflow for by object type (e.g. require object to reach drafted status, then review status, then approval status before allowing it to be flagged as ready for development)	1
31	R TS1.30 Create Glossary	Can create and update a glossary	1
32	R TS1.31 Upload Glossary	Can upload an existing glossary to the requirement management tool	1
33	R TS1.32 Assign owner of req	Can assign/re-assign a user to a single requirement or a set of requirements as the "owner"	3
34	R TS1.33 Receive notification	Owner receive notifications around requirements that are assigned to them (e.g. when it first gets assigned to them, when it's reassigned when it's edited, when it's approved)	1
35	R TS1.34 Create requirements plan	Can create/edit delivery iterations and assign requirements to them	2
36	R TS1.35 Visually see features	Can visually see features (or similar object type) as assigned to iterations (e.g. generate a roadmap for a release listing features assigned by iteration with ability to see progress)	2
37	⊞ R TS2 Requirements Elicitation	Requirements group for requirements elicitation	
41	⊞ R TS3 Requirements Analysis	Requirements group for requirements analysis	
78	⊞ R TS4 Requirements Specification	Requirements group for requirements specification	
100	⊞ R TS5 Requirements Validation	Requirements group for requirements validation	
116	⊞ R TS6 Requirements Management	Requirements group for requirements management	
155	⊞ R TS7 RM Tool User Experience	Requirements group for RM Tool user experience	
181	⊞ R TS8 RM Tool Administration	Requirements group for RM Tool administration	
205	⊞ R TS9 Others	Requirements group for others RM Tool requirements	

Fig. 3.10. Requirements Table for RM Tool Trade Study

Once a set of selection criteria with a defined priority level has been established, the weighting factors for each group of the selection criteria are calculated on the basis of the relative importance of each selection criterion group. In this case study, the AHP method is applied to calculate the weighting factors of the selection criteria group. The AHP method can also be used to calculate the weighting factors for each selection criterion separately, but since there are 207 selection criteria in total, the paired matrix would become very large. Thus, in this trade study, the AHP method is used only to determine the weighting factors of the selection criteria groups. The selection criteria have defined individual priority levels.

To calculate the weighting factors for each group of the selection criteria, a pairwise matrix is first constructed (Table 3.5). The level of importance of each group of the selection criteria is determined by judging the nine pairs of the groups according to the comparison scale provided in Table 2.16. The criteria groups in Table 3.5 are listed according to the id of the requirements group (Fig. 3.9).

Table 3.5. Pairwise matrix of trade study

#	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9
TS1	1	4.00	0.25	0.17	0.14	0.11	0.33	0.20	3.00
TS2	0.25	1	0.14	0.17	0.14	0.11	0.20	0.17	2.00
TS3	4.00	7.00	1	3.00	0.50	0.17	3.00	2.00	8.00
TS4	6.00	6.00	0.33	1	0.17	0.13	3.00	1.00	9.00
TS5	7.00	7.00	2.00	6.00	1	0.33	6.00	5.00	9.00
TS6	9.00	9.00	6.00	8.00	3.00	1	9.00	7.00	9.00
TS7	3.00	5.00	0.33	0.33	0.17	0.11	1	0.50	8.00
TS8	5.00	6.00	0.50	1.00	0.20	0.14	2.00	1	7.00
TS9	0.33	0.50	0.13	0.11	0.11	0.11	0.13	0.14	1
Total	**35.58**	**45.50**	**10.68**	**19.78**	**5.43**	**2.21**	**24.66**	**17.01**	**56.00**

The next step is to normalize the defined weights. First, each column entry is divided by the sum of the column (the ‘Total’ cell of a particular selection group) to obtain a normalized result (

Table **3.6**). Then, each row of the resulting matrix is summed up. Finally, the sums of the criteria are normalized with the objective to obtain the final weighting factor. Equations (11) and (12) show the weight normalization of the ‘TS1’ group of the selection criteria.

$$n_1 = \frac{v_1}{\sum_{i=1}^{9} v_1} = \frac{1}{35.58} = 0.03; \tag{11}$$

$$nw_1 = \frac{trv_1}{\sum_{i=1}^{9} trv_1} = \frac{0.30}{9.00} = 0.0337; \tag{12}$$

Where:

n1 – 1st criteria group normalized value by column,
nw1 – 1st criteria group weighting factor,
trv1 – 1st criteria group sum of the normalized value by row.

Table 3.6. Normalized Weights of Trade Study

#	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9	Total	Weighting Factor
TS1	0.03	0.09	0.02	0.01	0.03	0.05	0.01	0.01	0.05	**0.30**	**0.0337**
TS2	0.01	0.02	0.01	0.01	0.03	0.05	0.01	0.01	0.04	**0.18**	**0.0201**
TS3	0.11	0.15	0.09	0.15	0.09	0.08	0.12	0.12	0.14	**1.06**	**0.1179**
TS4	0.17	0.13	0.03	0.05	0.03	0.06	0.12	0.06	0.16	**0.81**	**0.0901**
TS5	0.20	0.15	0.19	0.30	0.18	0.15	0.24	0.29	0.16	**1.87**	**0.2082**
TS6	0.25	0.20	0.56	0.40	0.55	0.45	0.36	0.41	0.16	**3.36**	**0.3732**
TS7	0.08	0.11	0.03	0.02	0.03	0.05	0.04	0.03	0.14	**0.54**	**0.0596**
TS8	0.14	0.13	0.05	0.05	0.04	0.06	0.08	0.06	0.13	**0.74**	**0.0818**
TS9	0.01	0.01	0.01	0.01	0.02	0.05	0.01	0.01	0.02	**0.14**	**0.0155**
Total										**9.00**	**1.00**

The calculated weighting factors are stored for a specific selection criterion group by using the *normalizedPriority* tag of the *PrioritizedRequirement* stereotype.

In order to automate the trade study analysis in the MBSE environment, the *CapabilityConfiguration* element with a defined set of *measurements* is introduced. Fig. 3.11 (only a part of the figure is shown due to the actual size of the figure) shows the introduced 'RM Tool Configuration' *CapabilityConfiguration* that contains the required set of *measurements*. Each *measurement* represents the defined selection criterion of an RM tool trade study. *Measurements* are associated with a particular selection criterion requirement using the *Satisfy* relationship. Also, the *Criterion* stereotype is applied to *measurements* in order to store the priority level of the associated selection criterion requirement to the *weight* tag.

The final step of data preparation for the trade study process is to prepare a set of alternative RM solution architectures. In this trade study, eight alternatives are selected that represent the existing RM tools on the market. Fig. 3.12 (only a part of the figure is shown due to the actual size of the figure) shows alternatives that are defined as *FieldedCapability* elements typed by the 'RM Tool Configuration' *Capability Configuration.* Each *FieldedCapability* (RM_I, RM_II, RM_III, RM_IV, RM_V, RM_VI, RM_VII, RM_VIII) is evaluated and scored against the set of trade study selection criteria. Alternative RM solution architectures are scored by using the scale from 0 to 3 (0 – cannot do it in the tool; 1 – can do it but there is a manual workaround; 2 – can do it without any workarounds, but it is still not that easy to use; 3 – can do it in the tool, and it is user friendly). The complete list of alternatives with the determined scores of the selection criteria is provided in Appendix C.

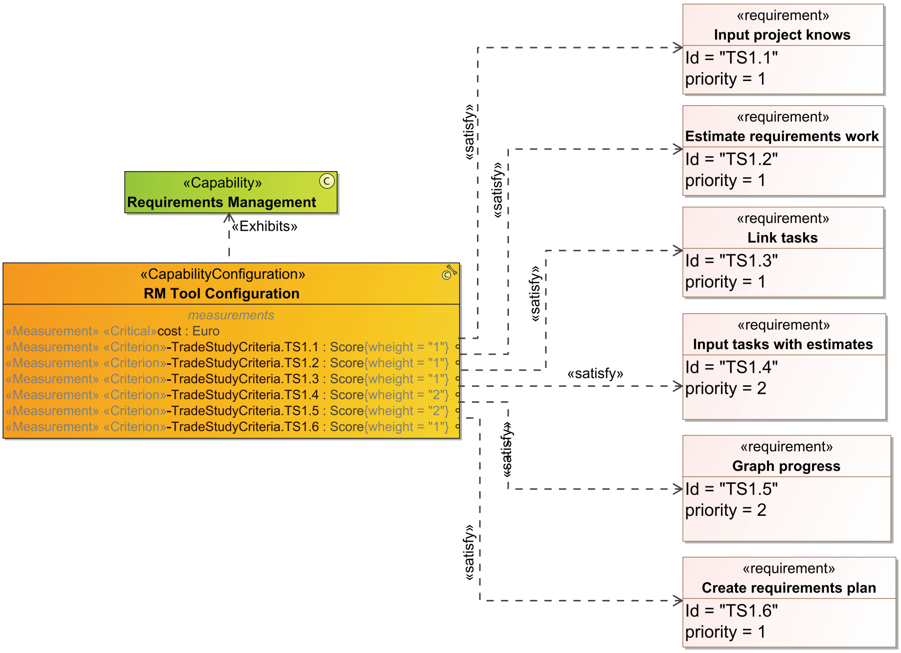

«Capability»
Requirements Management
«Exhibits»
«CapabilityConfiguration»
RM Tool Configuration
measurements
«Measurement» «Critical»cost : Euro
«Measurement» «Criterion»-TradeStudyCriteria.TS1.1 : Score{wheight = "1"}
«Measurement» «Criterion»-TradeStudyCriteria.TS1.2 : Score{wheight = "1"}
«Measurement» «Criterion»-TradeStudyCriteria.TS1.3 : Score{wheight = "1"}
«Measurement» «Criterion»-TradeStudyCriteria.TS1.4 : Score{wheight = "2"}
«Measurement» «Criterion»-TradeStudyCriteria.TS1.5 : Score{wheight = "2"}
«Measurement» «Criterion»-TradeStudyCriteria.TS1.6 : Score{wheight = "1"}
«satisfy»
«satisfy»
«satisfy»
«satisfy»
«satisfy»
«satisfy»
«requirement»
Input project knows
Id = "TS1.1"
priority = 1
«requirement»
Estimate requirements work
Id = "TS1.2"
priority = 1
«requirement»
Link tasks
Id = "TS1.3"
priority = 1
«requirement»
Input tasks with estimates
Id = "TS1.4"
priority = 2
«requirement»
Graph progress
Id = "TS1.5"
priority = 2
«requirement»
Create requirements plan
Id = "TS1.6"
priority = 1

Fig. 3.11. Configuration of the requirement management tool

The *ActualCriterion* stereotype is applied to *actual measurements* in order to store a calculated normalized score of a specific selection criterion during the deep check analysis. Furthermore, the *FieldedCapabilities* that represent alternative solution architectures of the RM tools are specified as the achieved configurations of ‘Enterprise X’ *ActualOrganization* using the *AchievedEffect* relationship (Fig. 3.12).

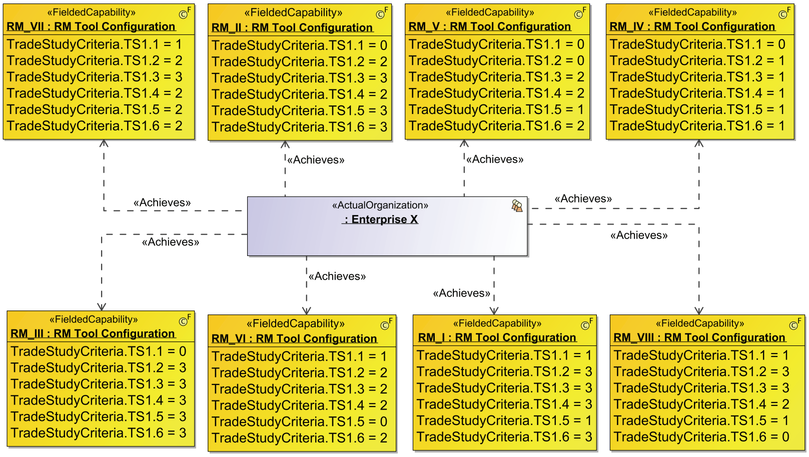

«FieldedCapability»
RM_VII : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 1
TradeStudyCriteria.TS1.2 = 2
TradeStudyCriteria.TS1.3 = 3
TradeStudyCriteria.TS1.4 = 2
TradeStudyCriteria.TS1.5 = 2
TradeStudyCriteria.TS1.6 = 2
«FieldedCapability»
RM_II : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 0
TradeStudyCriteria.TS1.2 = 2
TradeStudyCriteria.TS1.3 = 3
TradeStudyCriteria.TS1.4 = 2
TradeStudyCriteria.TS1.5 = 3
TradeStudyCriteria.TS1.6 = 3
«FieldedCapability»
RM_V : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 0
TradeStudyCriteria.TS1.2 = 0
TradeStudyCriteria.TS1.3 = 2
TradeStudyCriteria.TS1.4 = 2
TradeStudyCriteria.TS1.5 = 1
TradeStudyCriteria.TS1.6 = 2
«FieldedCapability»
RM_IV : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 0
TradeStudyCriteria.TS1.2 = 1
TradeStudyCriteria.TS1.3 = 1
TradeStudyCriteria.TS1.4 = 1
TradeStudyCriteria.TS1.5 = 1
TradeStudyCriteria.TS1.6 = 1
«Achieves»
«Achieves»
«Achieves»
«Achieves»
«ActualOrganization»
: Enterprise X
«Achieves»
«Achieves»
«Achieves»
«Achieves»
«FieldedCapability»
RM_III : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 0
TradeStudyCriteria.TS1.2 = 3
TradeStudyCriteria.TS1.3 = 3
TradeStudyCriteria.TS1.4 = 3
TradeStudyCriteria.TS1.5 = 3
TradeStudyCriteria.TS1.6 = 3
«FieldedCapability»
RM_VI : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 1
TradeStudyCriteria.TS1.2 = 2
TradeStudyCriteria.TS1.3 = 2
TradeStudyCriteria.TS1.4 = 2
TradeStudyCriteria.TS1.5 = 0
TradeStudyCriteria.TS1.6 = 2
«FieldedCapability»
RM_I : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 1
TradeStudyCriteria.TS1.2 = 3
TradeStudyCriteria.TS1.3 = 3
TradeStudyCriteria.TS1.4 = 3
TradeStudyCriteria.TS1.5 = 1
TradeStudyCriteria.TS1.6 = 3
«FieldedCapability»
RM_VIII : RM Tool Configuration
TradeStudyCriteria.TS1.1 = 1
TradeStudyCriteria.TS1.2 = 3
TradeStudyCriteria.TS1.3 = 3
TradeStudyCriteria.TS1.4 = 2
TradeStudyCriteria.TS1.5 = 1
TradeStudyCriteria.TS1.6 = 0

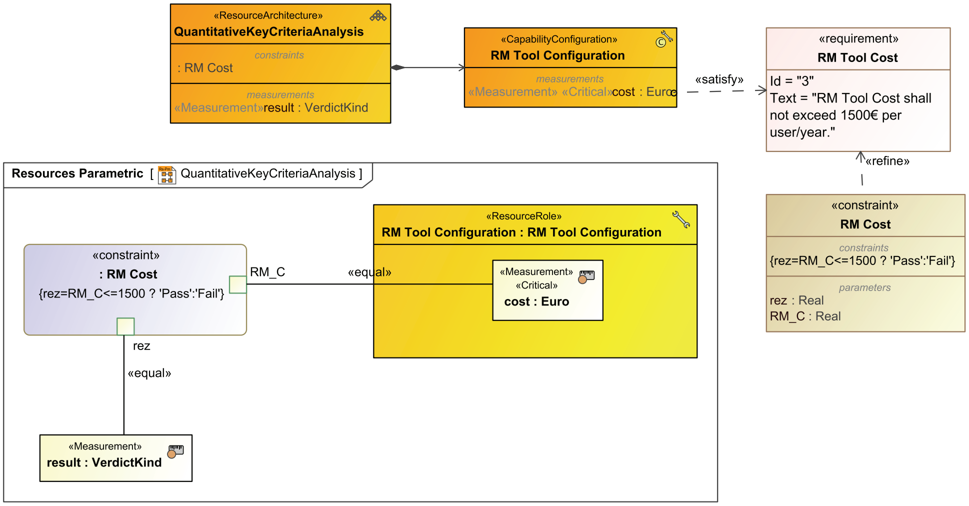
Fig. 3.12. Alternative solution architectures of the RM tool

Once the required data of trade study has been prepared, the next step is to perform the base check analysis.

Base check analysis: Check the coverage of key requirements (Automated step with simulation)

In order to check the coverage of the quantitative key requirements, first of all, it is necessary to formalize the text-based ‘RM Tool Cost’ *requirement*. The ‘RM Tool Cost’ requirement is refined by the ‘RM Cost’ *Constraint Block*, and *Constraint Parameters* are bounded to the *Measurements* that are verified by a *Requirement* (Fig. 3.13).

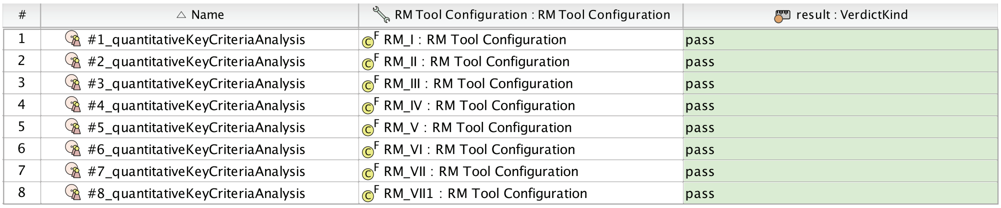
To automatize the analysis using the simulation, the *ResourceArchitecture* element ‘QuantitativeKeyCriteriaAnalysis’ is created, which will be executed during the simulation. Additionally, the *ResourceRole* typed ‘RM Tool Configuration’ *CapabilityConfiguration* is created, which links to alternative solution architectures. Finally, the ‘result’ *measurement* is introduced to store the analysis result.



«ResourceArchitecture»
QuantitativeKeyCriteriaAnalysis
constraints
: RM Cost
measurements
«Measurement»result : VerdictKind
«CapabilityConfiguration»
RM Tool Configuration
measurements
«Measurement» «Critical»cost : Euro
«satisfy»
«requirement»
RM Tool Cost
Id = "3"
Text = "RM Tool Cost shall not exceed 1500€ per user/year."
«refine»
«constraint»
RM Cost
constraints
{rez=RM_C<=1500 ? 'Pass':'Fail'}
parameters
rez : Real
RM_C : Real
Resources Parametric [QuantitativeKeyCriteriaAnalysis]
«ResourceRole»
RM Tool Configuration : RM Tool Configuration
«constraint»
: RM Cost
{rez=RM_C<=1500 ? 'Pass':'Fail'}
RM_C
«equal»
«Measurement»
«Critical»
cost : Euro
rez
«equal»
«Measurement»
result : VerdictKind

Fig. 3.13. Text-based requirement formalization

The coverage analysis of quantitative critical measurements is performed by simulating the ‘QuantitativeKeyCriteriaAnalysis’ with eight alternative solution architectures. Fig. 3.14 shows the results of the quantitative measurements coverage analysis.



#	△ Name	RM Tool Configuration : RM Tool Configuration	result : VerdictKind
1	#1_quantitativeKeyCriteriaAnalysis	RM_I : RM Tool Configuration	pass
2	#2_quantitativeKeyCriteriaAnalysis	RM_II : RM Tool Configuration	pass
3	#3_quantitativeKeyCriteriaAnalysis	RM_III : RM Tool Configuration	pass
4	#4_quantitativeKeyCriteriaAnalysis	RM_IV : RM Tool Configuration	pass
5	#5_quantitativeKeyCriteriaAnalysis	RM_V : RM Tool Configuration	pass
6	#6_quantitativeKeyCriteriaAnalysis	RM_VI : RM Tool Configuration	pass
7	#7_quantitativeKeyCriteriaAnalysis	RM_VII : RM Tool Configuration	pass
8	#8_quantitativeKeyCriteriaAnalysis	RM_VII1 : RM Tool Configuration	pass

Fig. 3.14. Results of key measurements coverage analysis in MSoSA tool

The next step is to check the coverage of the key capabilities and operational requirements. In order to do this, first of all, it is necessary to link each alternative RM architecture configuration to the critical *Capability* using the *Exhibits* relationship. Second, each *Resource Role* of an alternative RM architecture should be linked to the *Operational Role* using the *Implements* relationship. Finally, each *Resource Connector* of an alternative RM architecture should be linked to the *Operational Connector* using the *Implements* relationship (Fig. 3.15). The complete list of alternatives with traceability links between the critical interfaces is provided in Appendix A.

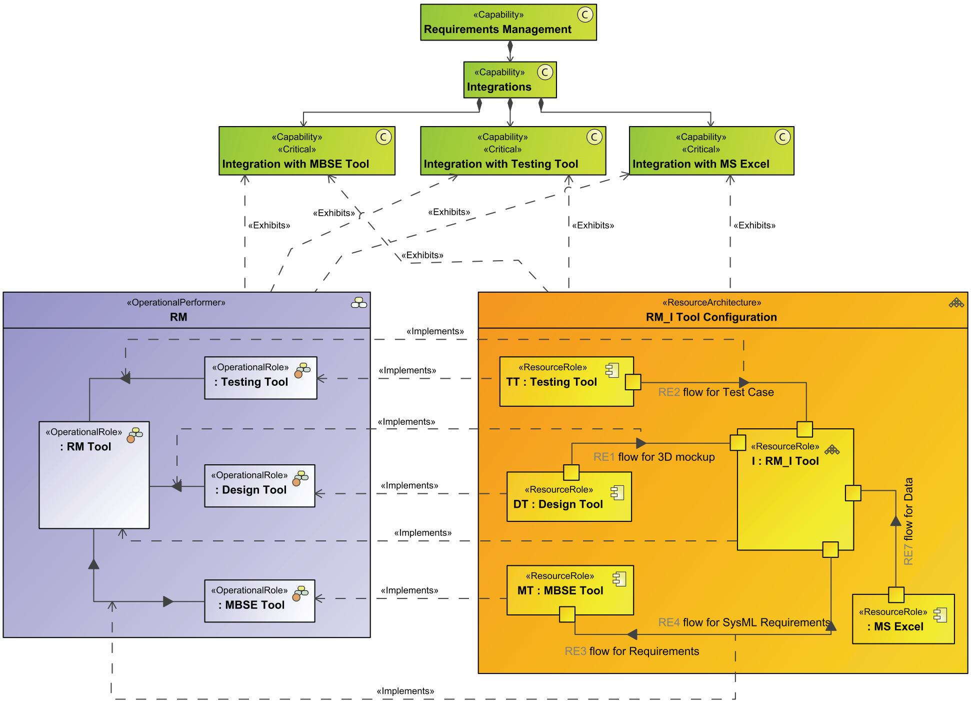
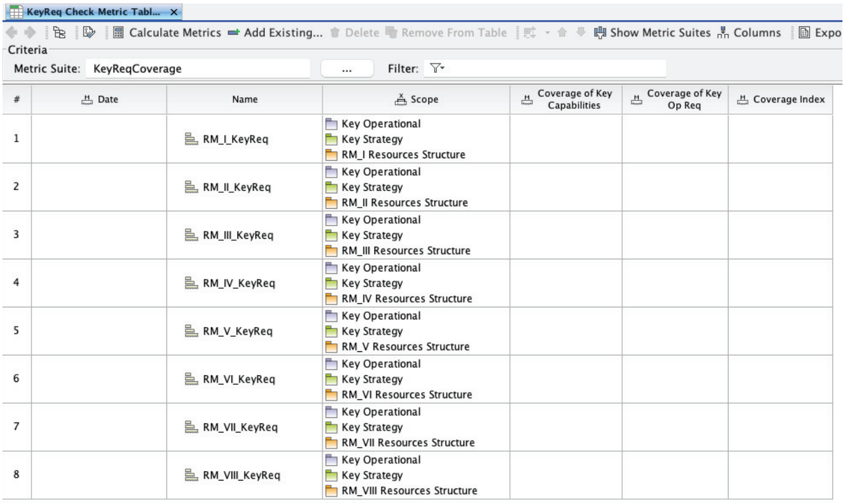


Fig. 3.15. Traceability links between critical interfaces and RM_I alternative

The specified metric suite with defined validation-based metric definitions (Fig. 3.2) is used to automate the analysis of the coverage of the key capabilities and operational requirements. Fig. 3.16 presents the metric table 'KeyReq Check Metric Table' that is created in the MSoSA tool to calculate coverage metrics for eight alternative SoS solution architectures. The 'KeyReqCoverage' metric suite is selected as a table metric suite.

In order to calculate the coverage metrics, first, for each alternative, an *InstanceSpecification* element typed by the 'KeyReqCoverage' metric suite is created. The results of the calculated metrics will be stored in the *InstanceSpecification* slots. Additionally, for each alternative, the scope of the metric is indicated: (1) package of key capabilities; (2) package of key operational requirements; (3) package of Rs domain views of a specific alternative.



Criteria

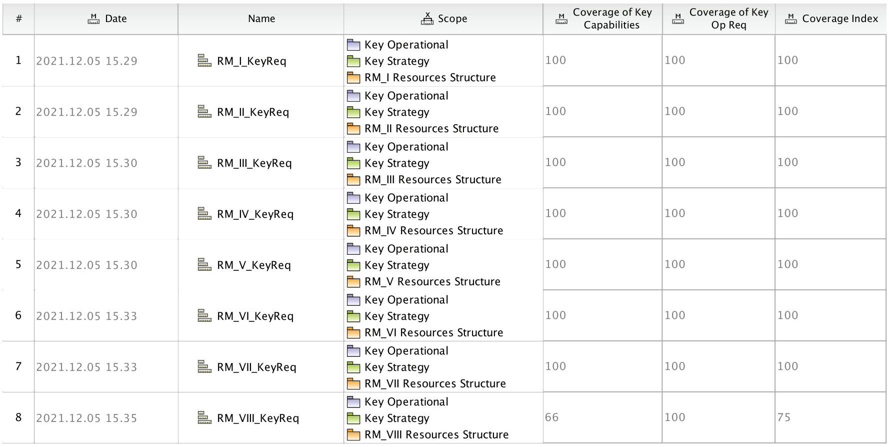
Metric Suite: KeyReqCoverage ...

Filter:

#	Date	Name	Scope	Coverage of Key Capabilities	Coverage of Key Op Req	Coverage Index
1		RM_I_KeyReq	Key Operational Key Strategy RM_I Resources Structure			
2		RM_II_KeyReq	Key Operational Key Strategy RM_II Resources Structure			
3		RM_III_KeyReq	Key Operational Key Strategy RM_III Resources Structure			
4		RM_IV_KeyReq	Key Operational Key Strategy RM_IV Resources Structure			
5		RM_V_KeyReq	Key Operational Key Strategy RM_V Resources Structure			
6		RM_VI_KeyReq	Key Operational Key Strategy RM_VI Resources Structure			
7		RM_VII_KeyReq	Key Operational Key Strategy RM_VII Resources Structure			
8		RM_VIII_KeyReq	Key Operational Key Strategy RM_VIII Resources Structure			

Fig. 3.16. Metric table for key requirements coverage analysis in MSoSA tool

Once the metric calculation begins, the calculated coverages in percentage values are provided for each alternative solution architecture (Fig. 3.17).



#	Date	Name	Scope	Coverage of Key Capabilities	Coverage of Key Op Req	Coverage Index
1	2021.12.05 15.29	RM_I_KeyReq	Key Operational Key Strategy RM_I Resources Structure	100	100	100
2	2021.12.05 15.29	RM_II_KeyReq	Key Operational Key Strategy RM_II Resources Structure	100	100	100
3	2021.12.05 15.30	RM_III_KeyReq	Key Operational Key Strategy RM_III Resources Structure	100	100	100
4	2021.12.05 15.30	RM_IV_KeyReq	Key Operational Key Strategy RM_IV Resources Structure	100	100	100
5	2021.12.05 15.30	RM_V_KeyReq	Key Operational Key Strategy RM_V Resources Structure	100	100	100
6	2021.12.05 15.33	RM_VI_KeyReq	Key Operational Key Strategy RM_VI Resources Structure	100	100	100
7	2021.12.05 15.33	RM_VII_KeyReq	Key Operational Key Strategy RM_VII Resources Structure	100	100	100
8	2021.12.05 15.35	RM_VIII_KeyReq	Key Operational Key Strategy RM_VIII Resources Structure	66	100	75

Fig. 3.17. Results of key requirements coverage analysis in the MSoSA tool

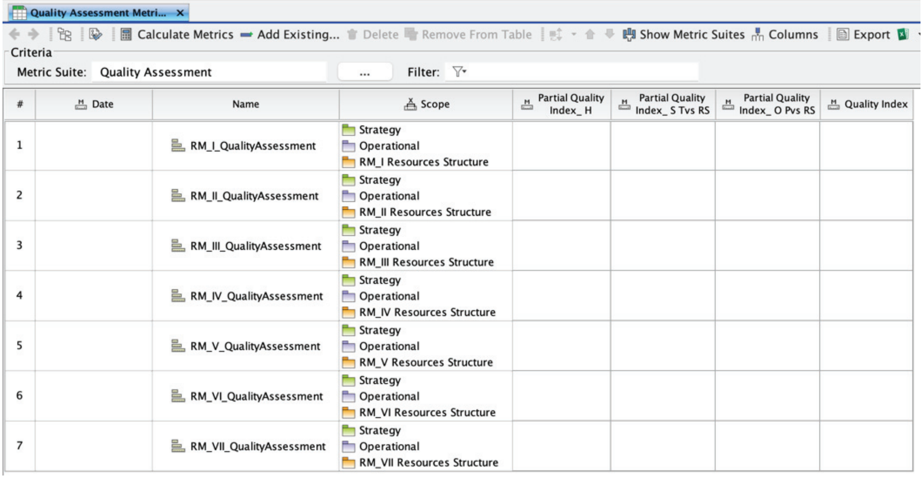
The column ‘Coverage of Key Capabilities’ shows the percentage coverage of *capabilities* that have applied the *Critical* stereotype. The column ‘Coverage of Key Op Req’ shows the percentage coverage of *OperationalActivities* that have applied the *Critical* stereotype. The column ‘Coverage Index’ shows the total coverage of the critical *capabilities* and *OperationalActivities*.

Once the coverage analysis of the quantitative key measurements, capabilities and operation activities is done, the results are reviewed. Fig. 3.14 demonstrates that all the checked alternatives have passed the quantitative measurements coverage check. Fig. 3.17 shows that the RM_VIII alternative does not fully cover the key capabilities. As the all identified key capabilities and operational requirements must be met, the alternative RM_VIII is excluded from the trade study.

Base check analysis: Quality assessment (Automated step with simulation)

The specified metric suite with defined validation-based metric definitions (Fig. 3.2) is used to automate the quality assessment. Fig. 3.18 presents the metric table 'Quality Assessment Metric Table' that is created in the MSoSA tool to calculate metrics for seven alternative solution architectures. The 'Quality Assessment' metric suite is selected as a table metric suite.

In order to calculate the metrics, first, for each alternative, an *InstanceSpecification* element typed by the 'Quality Assessment' metric suite is created. The results of the calculated metrics will be stored in the *InstanceSpecification* slots. Additionally, for each alternative, the scope of the metric is indicated: (1) package of St domain views; (2) package of Op domain views; (3) package of Rs domain views of a specific alternative.



Metric Suite: Quality Assessment ... Filter:

#	Date	Name	Scope	Partial Quality Index_ H	Partial Quality Index_ S Tvs RS	Partial Quality Index_ O Pvs RS	Quality Index
1		RM_I_QualityAssessment	Strategy Operational RM_I Resources Structure				
2		RM_II_QualityAssessment	Strategy Operational RM_II Resources Structure				
3		RM_III_QualityAssessment	Strategy Operational RM_III Resources Structure				
4		RM_IV_QualityAssessment	Strategy Operational RM_IV Resources Structure				
5		RM_V_QualityAssessment	Strategy Operational RM_V Resources Structure				
6		RM_VI_QualityAssessment	Strategy Operational RM_VI Resources Structure				
7		RM_VII_QualityAssessment	Strategy Operational RM_VII Resources Structure				

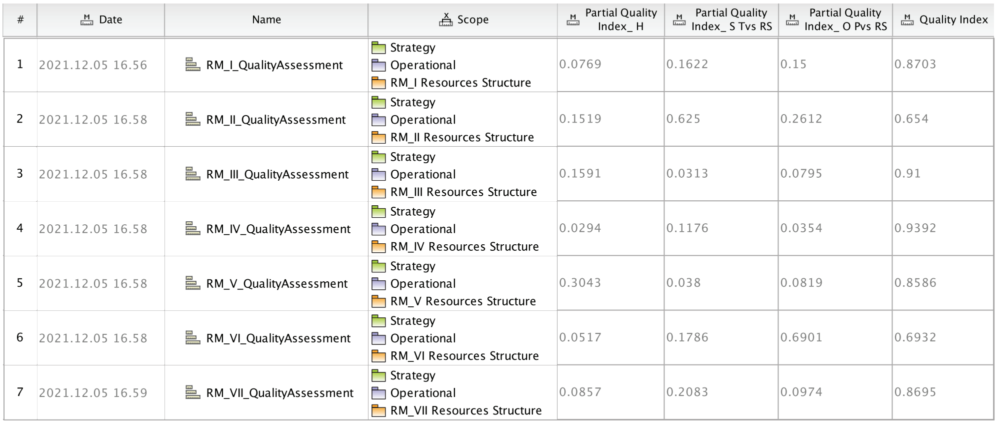
Fig. 3.18. Metric table for quality assessment in MSoSA tool

It is of importance to determine the minimum possible quality index before calculating the metrics. Those alternatives which do not reach the minimum values of the quality index will be excluded from the trade study due to poor quality. In this trade study, the minimum possible quality index is 0.7, as suggested by MIPS and CMS studies [136].

Once the metric calculation begins, each validation-based metric definition is calculated first, followed by partial quality index calculation based on a specific metric suite. Finally, the quality index is calculated by using Equation (4). The closer the quality index is to one, the higher is the quality of the evaluated alternative solution architecture.

Equation (13) shows the quality index calculation of the 'RM_I_QualityAssessment' *InstanceSpecification* that represents the RM_I solution architecture. The complete list of the quality index values for each alternative is provided in Fig. 3.19.

$$Q_{RM_I} = 1 - \left((0.0769 + 0.1622 + 0.15) \div 3\right) = 0.8703; \qquad (13)$$



#	Date	Name	Scope	Partial Quality Index_ H	Partial Quality Index_ S Tvs RS	Partial Quality Index_ O Pvs RS	Quality Index
1	2021.12.05 16.56	RM_I_QualityAssessment	Strategy Operational RM_I Resources Structure	0.0769	0.1622	0.15	0.8703
2	2021.12.05 16.58	RM_II_QualityAssessment	Strategy Operational RM_II Resources Structure	0.1519	0.625	0.2612	0.654
3	2021.12.05 16.58	RM_III_QualityAssessment	Strategy Operational RM_III Resources Structure	0.1591	0.0313	0.0795	0.91
4	2021.12.05 16.58	RM_IV_QualityAssessment	Strategy Operational RM_IV Resources Structure	0.0294	0.1176	0.0354	0.9392
5	2021.12.05 16.58	RM_V_QualityAssessment	Strategy Operational RM_V Resources Structure	0.3043	0.038	0.0819	0.8586
6	2021.12.05 16.58	RM_VI_QualityAssessment	Strategy Operational RM_VI Resources Structure	0.0517	0.1786	0.6901	0.6932
7	2021.12.05 16.59	RM_VII_QualityAssessment	Strategy Operational RM_VII Resources Structure	0.0857	0.2083	0.0974	0.8695

Fig. 3.19. Results of quality assessment in MSoSA tool

The column 'Partial Quality Index_H' presents the partial quality index by calculating the 'Horizontal' metric suite. The column 'Partial Quality Index_STvsRS' presents the partial quality index by calculating the 'St vs Rs' metric suite. The column 'Partial Quality Index_OPvsRS' presents the partial quality index by calculating the 'Op vs Rs' metric suite. The column 'Quality Index' presents the final quality index that indicates the quality level of the evaluated alternative solution architectures.

Once the metric calculation is done, the obtained results are reviewed. The RM_IV alternative has achieved the highest quality index value, specifically, 0.9392. However, two alternatives did not reach the specified minimum quality index value. Alternatives RM_II and RM_VI are thus excluded from the trade study, as their quality index is prominently lower than 0.7.

Deep check: Evaluation of alternative solution architectures by selection criteria (Automated step with simulation)

The next step is to perform the evaluation of alternative solution architectures by selection criteria. In order to automatize the analysis by using simulation, first, the *ResourceArchitecture* element 'RMTool_TradeStudyAnalysis' is created (Fig. 3.20), which will be executed during the simulation. Additionally, the *EvaluationByCriteria* stereotype is applied, and the *ResourceRole* typed 'RM Tool Configuration' *CapabilityConfiguration* is created, which contains the set of trade study selection criteria and links to alternative solution architectures.

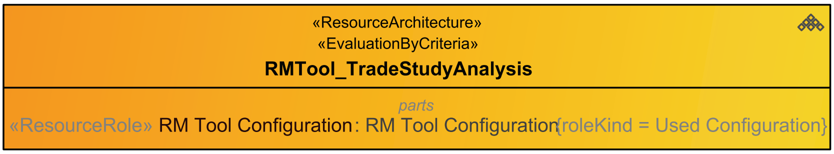
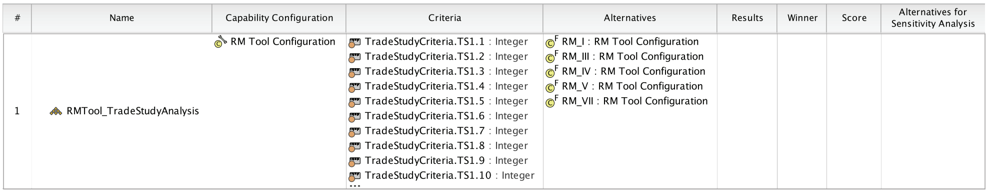

«ResourceArchitecture»
«EvaluationByCriteria»
RMTool_TradeStudyAnalysis
parts
«ResourceRole» RM Tool Configuration: RM Tool Configuration{roleKind = Used Configuration}

Fig. 3.20. Executable element for evaluation by selection criteria

Then, the dedicated *EvaluationByCriteriaTable* table is created to perform the architecture evaluation by selection criteria in the MSoSA tool (Fig. 3.21). The *ResourceArchitecture* element 'RMTool_TradeStudyAnalysis' (Fig. 3.20) is specified as a context of the table. The table columns ‘Criteria’ and ‘Alternatives’ display the derived data that is gathered based on the specified *ResourceRole* of the ‘RMTool_TradeStudyAnalysis’ *ResourceArchitecture*.



#	Name	Capability Configuration	Criteria	Alternatives	Results	Winner	Score	Alternatives for Sensitivity Analysis
1	RMTool_TradeStudyAnalysis	RM Tool Configuration	TradeStudyCriteria.TS1.1 : Integer TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.5 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.7 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.10 : Integer ...	RM_I : RM Tool Configuration RM_III : RM Tool Configuration RM_IV : RM Tool Configuration RM_V : RM Tool Configuration RM_VII : RM Tool Configuration				

Fig. 3.21. Table for Deep Check Analysis in MSoSA Tool

The following explains in detail the execution of the evaluation according to the algorithm provided in Fig. 2.23.

The deep check stage begins with an evaluation of each plausible RM tool alternative (RM_I, RM_III, RM_IV, RM_V, RM_VII) according to the established selection criteria. Plausible alternatives are judged by their overall satisfaction with a series of desirable characteristics by executing the model using the algorithm shown in Fig. 2.23.

First, the evaluation starts by normalizing the values of the scored criteria for each plausible alternative by the selection criterion. Equation (14) shows the normalization of the ‘TradeStudyCriterion.TS1.3’ selection criterion of the RM_I alternative. The complete list of normalized values of the scored criteria for each plausible alternative is provided in Appendix D.

$$nv_3 = \frac{v_3}{\sum_{i=1}^{5} v_3} = \frac{3}{3+3+1+2+3} = 0.25; \tag{14}$$

Second, the weighted scores are calculated. Normalized values of the selection criteria are multiplied by a priority level of a particular selection criterion as well as by a weighted factor of the relevant selection criteria group. Equation (15) shows the weighted score calculation of the ‘TradeStudyCriterion.TS1.3’ selection criterion of the RM_I alternative. The complete list of weighted values of the selection criteria for each plausible alternative is provided in Appendix D.

$$ws_3 = 0.25 \times 1 \times 0.0337 = 0.0084; \tag{15}$$

Third, the weighted scores are summed up for each alternative by using Equation (9). Table 3.7 shows the weighted scores summed by the selection criteria group, including the total sum of the weighted score of all the selection criteria.

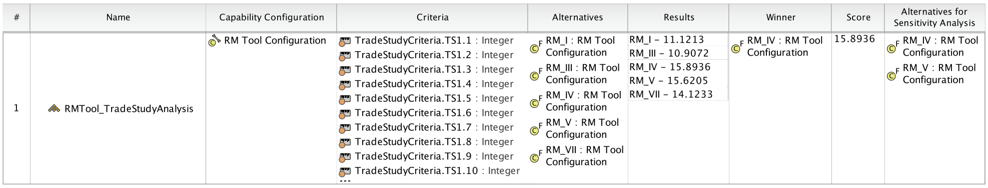
Table 3.7. Final scores of the evaluation by selection criteria

Criteria group	Sum of weighted scores				
	RM_I	RM_III	RM_IV	RM_V	RM_VII
TS1	0.4561	0.4334	0.5268	0.5277	0.5498
TS2	0.0146	0.0255	0.0179	0.0278	0.0146
TS3	2.5223	1.4470	3.4475	3.3564	3.3748

Criteria group	Sum of weighted scores				
	RM_I	RM_III	RM_IV	RM_V	RM_VII
TS4	1.2126	0.9657	1.4078	1.3037	1.5074
TS5	1.3980	1.2689	1.3446	1.5783	1.2808
TS6	3.8831	4.7714	7.0407	6.8397	5.8282
TS7	0.6399	0.8377	0.9628	0.8195	0.7332
TS8	0.9261	1.0968	1.0924	1.1008	0.7737
TS9	0.0686	0.0608	0.0531	0.0666	0.0608

Sum:	**11.1213**	**10.9072**	**15.8936**	**15.6205**	**14.1233**
Final Score:	**69.97**	**68.63**	**100**	**98.28**	**88.86**

Fig. 3.22 shows the results of an evaluation by the selection criteria that is entered in the MSoSA tool after the execution of the ‘RMTool_TradeStudyAnalysis’ element. The ‘Results’ column provides the final score for all the evaluated alternative solution architectures: RM_I – 11.1213; RM_III – 10.9072; RM_IV – 15.8936; RM_V – 15.6205; RM_VII – 14.1233. The ‘Winner’ column indicates the RM_IV alternative as a ‘winner’ as it has achieved the highest score of all the alternatives. The ‘Score’ column shows the reach score of the ‘winner’ alternative, that is, 15.8936. The ‘Alternatives for Sensitivity Analysis’ column indicates the alternatives that have obtained the same or similar final score: RM_IV and RM_V. The alternatives that scored 95% or higher in comparison to the highest-scoring alternative are suggested for sensitivity analysis including the highest-scoring alternative itself.



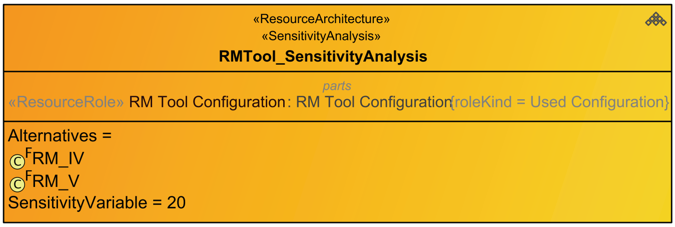
#	Name	Capability Configuration	Criteria	Alternatives	Results	Winner	Score	Alternatives for Sensitivity Analysis
1	RMTool_TradeStudyAnalysis	RM Tool Configuration	TradeStudyCriteria.TS1.1 : Integer TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.5 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.7 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.10 : Integer ...	RM_I : RM Tool Configuration RM_III : RM Tool Configuration RM_IV : RM Tool Configuration RM_V : RM Tool Configuration RM_VII : RM Tool Configuration	RM_I – 11.1213 RM_III – 10.9072 RM_IV – 15.8936 RM_V – 15.6205 RM_VII – 14.1233	RM_IV : RM Tool Configuration	15.8936	RM_IV : RM Tool Configuration RM_V : RM Tool Configuration

Fig. 3.22. Results of an evaluation using selection criteria in MSoSA tool

In order to ensure that the ‘winning’ alternative is truly the winner, it is necessary to perform a sensitivity analysis for two alternatives: RM_IV and RM_V.

Deep check: Sensitivity analysis (Automated step with simulation)

Once the evaluation by the selection criteria has been performed and there are alternatives that have a close final score, the sensitivity analysis is required. To automatize the sensitivity analysis using the simulation, the *ResourceArchitecture* element ‘RMTool_SensitivityAnalysis’ is created (Fig. 3.23), which will be executed during the simulation. The *SensitivityAnalysis* stereotype is applied to the ‘RMTool_SensitivityAnalysis’ element in order to specify the alternatives that will be analyzed – RM_IV and RM_V – and the sensitivity variable ‘20’. Additionally, the *ResourceRole* typed ‘RM Tool Configuration’ *CapabilityConfiguration* is created which contains the set of selection criteria that are needed during the sensitivity analysis.



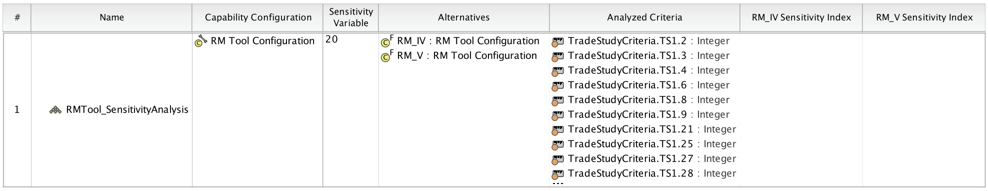
«ResourceArchitecture»
«SensitivityAnalysis»
RMTool_SensitivityAnalysis

parts
«ResourceRole» RM Tool Configuration : RM Tool Configuration{roleKind = Used Configuration}

Alternatives =
RM_IV
RM_V
SensitivityVariable = 20

Fig. 3.23. Executable element for sensitivity analysis

Then, the dedicated *SensitivityAnalysisTable* table is created to execute the sensitivity analysis in the MSoSA tool (Fig. 3.24). The *ResourceArchitecture* element 'RMTool_SensitivityAnalysis' (Fig. 3.23) is specified as a context of the table. The table column 'Analyzed Criteria' displays the derived data that are collected based on the specified *ResourceRole* of the 'RMTool_SensitivityAnalysis' *ResourceArchitecture*. This column provides a list of selection criteria which have a different actual score between the RM_IV and RM_V alternatives. Only the selection criteria with different scores are included in the sensitivity analysis, as changing the criteria with the same score will lead to the same change.



#	Name	Capability Configuration	Sensitivity Variable	Alternatives	Analyzed Criteria	RM_IV Sensitivity Index	RM_V Sensitivity Index
1	RMTool_SensitivityAnalysis	RM Tool Configuration	20	RM_IV : RM Tool Configuration RM_V : RM Tool Configuration	TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.21 : Integer TradeStudyCriteria.TS1.25 : Integer TradeStudyCriteria.TS1.27 : Integer TradeStudyCriteria.TS1.28 : Integer ...		

Fig. 3.24. Table for sensitivity analysis in MSoSA tool

The following explains in detail the execution of the sensitivity analysis according to the algorithm provided in Fig. 2.25.

The sensitivity analysis begins with the calculation of the difference between the actual score and the actual score increased by the sensitivity variable. Equation (16) shows the calculation of the difference of the actual score of the RM_IV alternative selection criterion 'TradeStudyCriterion.TS6.22' when the sensitivity variable is equal to 20%.

$$\Delta x_{22} = (3 \times 120\%) - 3 = 0.60; \tag{16}$$

The next step is to calculate the difference in the final score after the sensitivity variable increases one selection criterion. This step consists of multiple calculations. First, the increased actual score criterion is normalized. Equation (17) shows the normalization of the 'TradeStudyCriterion.TS6.22' selection criterion of the RM_IV alternative.

$$nv_{22} = \frac{iv_{22}}{\sum_{i=1}^{2} v_{22}} = \frac{3}{3+2} = 0.6; \tag{17}$$

Second, the weighted score is calculated. The normalized value of the selection criterion is multiplied by a priority level of a particular selection criterion and by a weighted factor of the relevant selection criterion group. Equation (18) shows the weighted score calculation of the ‘TradeStudyCriterion.TS6.22’ selection criterion of the RM_IV alternative.

$$ws_{22} = 0.3732 \times 2 \times 0.6 = 0.4478; \qquad (18)$$

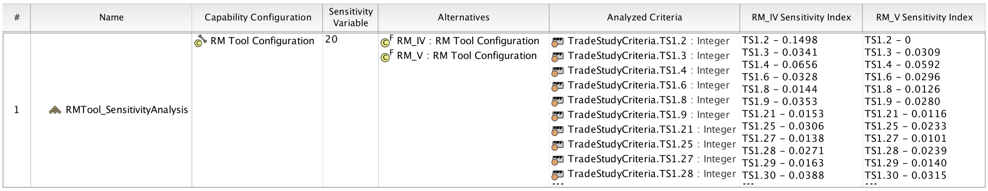
Third, the newly calculated weighted score is summed up with the weighted scores of the other criteria calculated in the deep check analysis. Fourth, the difference in the final score is calculated after the sensitivity variable increases one selection criterion. Equation (19) shows the difference calculation of the final score when the ‘TradeStudyCriterion.TS6.22’ selection criterion of the RM_IV alternative is changed by the sensitivity variable.

$$\Delta y_{22} = 4.8693 - 4.6658 = 0.2036; \qquad (19)$$

The final step in the sensitivity analysis is to calculate the sensitivity index. Equation (20) shows the calculation of the sensitivity index of the selection criterion ‘TradeStudyCriterion.TS6.22’ of the RM_IV alternative. The complete list of calculations of the sensitivity indexes is provided in Appendix E.

$$sensitivity_{22} = \frac{0.2036}{0.60} = 0.3393; \qquad (20)$$

Fig. 3.25 presents the sensitivity analysis results that are provided in the MSoSA tool after the execution of the ‘RMTool_SensitivityAnalysis’ element. The ‘RM_IV Sensitivity index’ column provides calculated sensitivity indexes of the RM_IV alternative for the selection criteria shown in the ‘Analyzed Criteria’ column. The ‘RM_V Sensitivity index’ column provides calculated sensitivity indexes of the RM_V alternative. The higher is the sensitivity index value, the more sensitive is a particular selection criterion.



#	Name	Capability Configuration	Sensitivity Variable	Alternatives	Analyzed Criteria	RM_IV Sensitivity Index	RM_V Sensitivity Index
1	RMTool_SensitivityAnalysis	RM Tool Configuration	20	RM_IV : RM Tool Configuration RM_V : RM Tool Configuration	TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.21 : Integer TradeStudyCriteria.TS1.25 : Integer TradeStudyCriteria.TS1.27 : Integer TradeStudyCriteria.TS1.28 : Integer ---	TS1.2 – 0.1498 TS1.3 – 0.0341 TS1.4 – 0.0656 TS1.6 – 0.0328 TS1.8 – 0.0144 TS1.9 – 0.0353 TS1.21 – 0.0153 TS1.25 – 0.0306 TS1.27 – 0.0138 TS1.28 – 0.0271 TS1.29 – 0.0163 TS1.30 – 0.0388 ---	TS1.2 – 0 TS1.3 – 0.0309 TS1.4 – 0.0592 TS1.6 – 0.0296 TS1.8 – 0.0126 TS1.9 – 0.0280 TS1.21 – 0.0116 TS1.25 – 0.0233 TS1.27 – 0.0101 TS1.28 – 0.0239 TS1.29 – 0.0140 TS1.30 – 0.0315 ---

Fig. 3.25. Sensitivity analysis results in MSoSA tool

The ten most sensitive selection criteria are selected for further analysis according to the calculated sensitivity indexes of the RM_IV and RM_V alternatives (Table 3.8).

Table 3.8. Ten most sensitive selection criteria after sensitivity analysis

#	Cri.	Prior.	Sensitivity Variable +20%							
			RM_IV				RM_V			
			Act Value	Δx	Δy	S index	Act Value	Δx	Δy	S index
1	**TS3.8**	8	1	0.2	0.0938	0.4688	3	0.6	0.2515	0.4192
2	**TS3.9**	8	3	0.6	0.2315	0.3859	2	0.4	0.1132	0.2830
3	**TS3.10**	8	1	0.2	0.0938	0.4688	2	0.6	0.2515	0.4192
4	**TS3.18**	2	2	0.4	0.1415	0.3537	0	0	0.0000	0.0000
5	**TS3.22**	8	1.5	0.3	0.2001	0.6669	1	0.2	0.0922	0.4611
6	**TS5.1**	2	1	0.2	0.0556	0.2782	3	0.6	0.1537	0.2562
7	**TS6.6**	2	1.5	0.3	0.1583	0.5278	1	0.2	0.0730	0.3649
8	**TS6.7**	2	2	0.4	0.1448	0.3620	3	0.6	0.1860	0.3100
9	**TS6.22**	2	3	0.6	0.2036	0.3393	2	0.4	0.1031	0.2578
10	**TS6.31**	2	3	0.6	0.2205	0.3675	2	0.4	0.1144	0.2861

Once the sensitivity analysis has been performed, the results should be reviewed, and the final decision should be made thereby indicating the 'winner' alternative solution architecture.

Results review

In order to make the final decision, Table 3.8 data is reviewed. Four selection criteria (TS3.8; TS3.9; TS3.10; TS3.22) have the highest priority (specifically, 8), which means that the selection criteria are essential for the tool to be effective in managing the requirements. The weighted score of these four criteria is calculated to compare which of the alternatives is superior. Equation (21) shows the weighted score calculation of RM_IV. Equation (22) shows the weighted scored calculation of RM_V.

$$ws_{IV} = 0.1179 \times 8 \times (1 + 3 + 1 + 1{,}5) = 6.1308; \tag{21}$$

$$ws_{V} = 0.1179 \times 8 \times (3 + 2 + 2 + 1) = 7.5456; \tag{22}$$

Calculations (21) and (22) revealed that the RM_V alternative is superior to RM_IV. Therefore, the final verdict after the sensitivity analysis is that the alternative tool RM_V is selected as the 'winner' RM tool for Enterprise X.

3.2.3 Reliability and validity check of the experiment result

In this section, the reliability and validity check of the experiment result is presented.

Reliability

A reliability check was carried out by repeating the RM tool trade study five times, which is presented in Section 3.2.2, each time removing one or more alternatives from the original set of alternatives (Table 3.9). The main purpose of the reliability check is to ensure that the results of the alternative estimates are the same regardless of the set of alternatives being evaluated.

Table 3.9. Results of reliability check of experimental results

		I	II	III	IV	V
Removed alternatives		RM_V	RM_II RM_VI	RM_V RM_VI RM_VII	RM_I RM_II RM_III	RM_IV
Quality Index	**RM_I**	0.8703	0.8703	0.8703	-	0.8703
	RM_II	0.654	-	0.654	-	0.654
	RM_III	0.91	0.91	0.91	-	0.91
	RM_IV	0.9392	0.9392	0.9392	0.9392	-
	RM_V	-	0.8586	-	0.8586	0.8586
	RM_VI	0.6932	-	-	0.6932	0.6932
	RM_VII	0.8695	0.8695	-	0.8695	-
Results by selection criteria	**RM_I**	11.1213	11.1213	11.1213	-	11.1213
	RM_III	10.9072	10.9072	10.9072	-	10.9072
	RM_IV	15.8936	15.8936	15.8936	15.8936	-
	RM_V	-	15.6205	-	15.6205	15.6205
	RM_VII	14.1233	14.1233	-	14.1233	-
Are the results the same as the case study of RM tool?		Yes	Yes	Yes	Yes	Yes

The conducted reliability check of the experiment result revealed that the same evaluation results were obtained regardless of the original set of alternatives that were included in the trade study. Thus, the conducted check confirmed that the experiment result is reliable.

Validity

A validity check of the experiment was carried out by applying the UT3SA method to three experiments: electric road configuration (ERC), marine search and rescue study (MSR), and RM (Table 3.10). The results of the experiments which were carried out by using the UT3SA method were evaluated by an expert group (Section 3.2) and were based on published researches. The main purpose of the validity check is to ensure that the UT3SA method selects the most preferred architecture alternative from the available alternatives.

Table 3.10. Results of validity check of experimental results

	ERC	MSR	RM
Source of results	Experts group Previous researches [142] [52]	Experts group	Experts group Previous researches [143] [144]

	ERC	MSR	RM
Expected vs actual results	ERC_I/ERC_I	MSR_II/MSR_II	RM_V/RM_V
Are the expected and actual results the same?	Yes	Yes	Yes

The conducted validity check of the experiment results revealed that the actual evaluation results were obtained as expected. The expected results were determined by an expert group and were based on published researches. Thus, the validity check confirmed that the experiment result is valid.

3.2.4 Evaluation of UT3SA quality improvement

The introduced UT3SA method seeks to increase the quality of a trade study by providing means to evaluate alternative SoS architectures in the MBSE environment, which is aimed at automating the evaluation of SoS architectures by using model execution. Trade study quality improvement refers to the reduction of potential errors in evaluating alternative SoS architectures. It follows the reduction of the risk of choosing an inappropriate alternative SoS architecture.

The UT3SA method includes the following means: MBSE, UAF, automation by simulation. There are many published researches that have already proven and confirmed that these means improve the quality by reducing the number of possible errors:

- MBSE – as per [145] – provides a review of the existing literature that summarizes the benefits of MBSE in the SE literature. After reviewing 360 papers, a list of the MBSE benefits was compiled, and, among the most frequently mentioned benefits, the following points can be outlined: better system quality, error reduction, and improved system design. From the quality improvement perspective, there are 127 articles that state this as an advantage of MBSE. Source [65] reveals how MBSE assisted in the Europa Clipper Project at NASA's JPL. JPL stated that MBSE "provides stable, repeatable inputs for driving these analyses, reducing errors, increasing confidence, and dramatically decreasing cycle time" [65]. In addition, they stated that the "System Model has improved communications among members of the flight project resulting in saved time, reduction in errors, and reduction in drudge work" [65].
- UAF – as per [146] – presents that the National Environmental Satellite, Data, and Information Service (NESDIS) ground enterprise and NASA's JPL ground project are successful examples of SoS architectures designed with UAF. The study showed that MBSE along with UAF reduces rework, increases accuracy, reduces errors, and allows robust and informed decision-making. Source [147] presents the US DoD research projects that describe how the use of UAF eliminates

the limitations of solving problems, improves quality and reduces errors.

- What concerns automation by simulation – as per [148], [149], [150], [151] papers – it can be stated that simulation improves quality and helps reduce errors. Additionally, the *Deloitte* survey [152] revealed that 92% of their respondents said that automation met or exceeded their expectations. This was followed by improved quality/accuracy (90%), improved productivity (86%), and cost reduction (59%).

Additionally, a study was conducted to compare the number of errors found by automated and manual evaluation of the SoS architecture. The study was divided into two evaluations: Base Check and Deep Check. During the Base Check, the number of errors detected when checking SoS architectures was determined by applying the evaluation principles of the Base Check step of the UT3SA method. During the Deep Check, the number of errors made in assessing the compliance of trade study selection criteria was determined by applying the evaluation principles of the Deep Check step of the UT3SA method. Automated evaluation was performed by using the automated means provided by the UT3SA method based on simulation, while manual evaluation was performed by a group of experts (Table 3.11).

Table 3.11. Comparison of the number of errors found during automated and manual evaluations

		Project I	**Project II**	**Project III**
Number of elements in SoS architecture		352241	57628	108558
Base Check: Number of errors found in SoS architectures	**Automated evaluation**	278	27	66
	Manual evaluation	171	21	47
	Difference in %	38.49%	22.22%	28.79%
Deep Check: Number of errors made in assessing selection criteria	**Automated evaluation**	0	0	0
	Manual evaluation	23	0	6

The study revealed that using automated means in the Base Check step detects at least 22% more errors than the manual evaluation of SoS architectures. Furthermore, the study disclosed that the occurrence of errors is inevitable when evaluating the compliance of trade study criteria manually. Meanwhile, automated evaluation eliminates errors caused by the human factor.

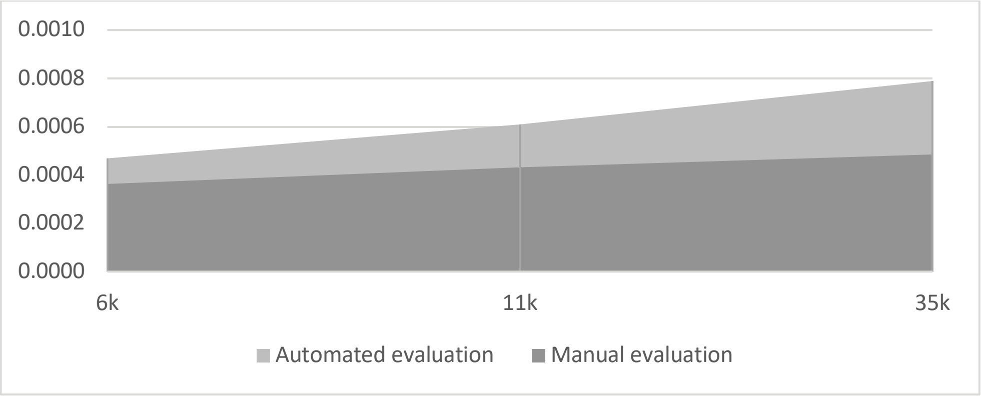


Fig. 3.26. Ratio of found errors and the number of elements in SoS architecture

However, the number of the errors found is closely related to the number of the evaluated elements in the SoS architecture models. Fig. 3.26 shows the relationship between the number of the errors found and the number of elements in the SoS architecture. The higher is the number of elements, the more likely it is to find errors in those models.

3.3 Experimental Evaluation Conclusions

The conclusions of this chapter are as follows.

1. Conclusions on the case study conducted:
   a. The coverage analysis of the key capabilities, operational requirements and quantified measurements at the ‘Base Check’ stage showed that the RM_VIII alternative solution architecture does not meet all the critical requirements. Therefore, the RM_VIII alternative was excluded from the further stages of the trade study. The coverage analysis demonstrates its importance in the trade study: (1) it helps to verify the compliance of the critical requirements in an early phase of the trade study; (2) it eliminates inappropriate alternatives, thereby reducing the number of alternatives in a study, which speeds up the analysis itself; (3) it reduces the risk of choosing an inappropriate alternative in the later stages that does not meet the critical requirements.
   b. The quality assessment at the ‘Base Check’ stage showed that the RM_II and RM_VI alternative solution architectures do not reach the specified minimum quality index value of 0.7. Thus, the RM_II and RM_VI alternatives were excluded from the trade study because of their poor quality. The quality assessment demonstrates its importance as it evaluates alternative solution architectures in the early phase of the trade study. Therefore, in the further steps of the trade study, the evaluation is performed only with a set of plausible solution architectures. The second and

    third benefits of the first conclusion also apply to quality assessment.

    c. The results of the evaluation of alternative solution architectures by the selection criteria revealed that the RM_IV solution architecture reached the highest score. However, the RM_V solution architecture reached a very close score (with a difference of <=5%) to that of RM_IV. In such a case, a final decision cannot be made yet, and a sensitivity analysis is required to verify the selection of the conclusive choice of the preferred solution architecture.
    d. A review of the sensitivity analysis results and the priority levels of the selection criteria revealed that the RM_IV solution architecture, which achieved the highest score after the evaluation by the selection criteria, is not superior to the RM_V alternative, which achieved a slightly lower score than RM_IV. The RM_V alternative is chosen as the 'winner' solution architecture of the RM tool trade study as it meets more critical selection criteria than RM_IV. The sensitivity analysis demonstrates its importance in a trade study as it helps to clarify the choice of the preferred solution architecture among very similar ones based on the importance of the selection criteria.
2. Conclusions on the reliability and validity check of the experiment result:
    a. The reliability check carried out with the experiment result revealed and confirmed that the experiment result is reliable since the same evaluation results were obtained by running five evaluations regardless of the original set of alternatives that were included in the trade study.
    b. The conducted validity check of the experiment result revealed and confirmed that the experiment result is valid since the evaluation results determined by the expert group and previous studies were consistent with the results determined using the UT3SA method.
3. Conclusions on the evaluation of the quality improvement of UT3SA in the trade study are as follows:
    a. The analysis of how and whether MBSE, UAF and automation by simulation contribute to quality improvement in the trade study revealed that these means have a significant contribution to quality improvement by reducing the number of errors. This statement is supported by many researchers from academia, SE, and industry. Additionally, *Deloitte*'s survey revealed that 92% of the respondents confirmed that automation and simulation met or exceeded their expectations, and their application resulted in improved quality/accuracy (90%), improved productivity (86%) and reduced costs (59%).

b. The conducted study which compared the number of errors detected by automated (performed by automated means of the UT3SA method) and manual (performed by an expert group) evaluation of the SoS architecture revealed that automated evaluation detects at least 22% more errors. In addition, the number of errors depends on the size of the model. The higher is the number of elements, the higher is the probability of finding errors.

4 CONCLUSIONS

1. Analysis of ADFs revealed that the already existing ADFs do not define how a trade study for SoS architectures should be conducted, nor do they provide a formalized trade study process. ADFs provide only a basis for collecting the data required for trade studies from the architecture point of view, and enable standard-based execution of SoS architecture models with some limitations. However, no concepts are provided that are needed to specify the specific trade study data.
2. Analysis of MBSE, UAF, and fUML revealed that they are a suitable means for automating evaluations of alternative SoS architectures in the trade study process. However, extensions of the UAFP metamodel are necessary to perform a trade study in the MBSE environment, as it lacks specific concepts to define and perform a trade study in MBSE, along with UAF and fUML.
3. Analysis of the trade study processes showed that the majority of the already existing trade study processes: (1) do not provide the formally defined trade study process; (2) define only the main flow of a trade study excluding details of alternative evaluation, such as specific evaluation methods, techniques with algorithms or constraints; (3) are not adapted to the domain; therefore, they provide only general steps and use general terminology; (4) do not provide any guidance how to apply the trade study process in the MBSE environment; (5) do not define the roles or the input/output data for each process step. As UAF is an OMG standard that encompasses all the necessary characteristics to build architecture descriptions of software-intensive 21st century systems, it can be used as a basis for the formalization and extension of a trade study process for SoS architectures. This could help alleviate deficiencies in the already existing trade study processes. Additionally, UAF enables the usage of simulation-based methods to evaluate alternative solution architectures.
4. Based on the analysis, a new method for the trade study is developed. The proposed method (UT3SA) is currently the only known method that integrates all the following characteristics: (1) a formally defined trade study process with defined input/output data and automated steps; (2) guidance on how to define the trade study in the MBSE environment; (3) detailed architecture evaluation methods within the trade study process with algorithms and constraints; (4) support of early architecture quality assessment.
5. The UT3SA method is implemented as a UAF profile with the extensions related to trade studies (the UT3SA profile) and a package of validation rules, validation-based metric definitions, and tables (UT3SA constraints). The application of the UT3SA method allows defining the required data of a trade study directly in the model, assess the quality of the alternative solution architectures at an early stage of the trade study, evaluate the alternative solution architectures according to the established

selection criteria, and conduct sensitivity analysis to verify the selection of the preferred solution architecture. The UT3SA profile and the UT3SA constraints package are developed by using the MSoSA v2021xRefresh2 CASE tool, and they can be used in any compatible UAF CASE tool.

6. The UT3SA method is applied to an RM tool trade study with eight alternative solution architectures to evaluate the suitability of the proposed method. Four evaluations are performed during the trade study: (1) coverage check of the key requirements; (2) quality assessment; (3) evaluation of the alternative solution architectures by the selection criteria; (4) sensitivity analysis. The experiment proved that the UT3SA method provides: (1) the necessary extensions of the UAF profile, which allows defining directly the trade study in a model; (2) a detailed trade study process covering specific steps, input/output data, and evaluation methods that led to a comprehensive trade study; (3) the necessary set of constraints and algorithms for the automated evaluation of architectures so that to select the most appropriate solution architecture from several possible alternatives. Furthermore, the experimental evaluation confirmed that the proposed UT3SA method is applicable in practice.
7. The performed reliability and validity check of the experiment result revealed and confirmed that the experiment result is reliable and valid. Therefore, the UT3SA method selects the preferred SoS architecture alternative from the available alternatives.
8. The evaluation of the UT3SA quality improvement conducted in a trade study revealed that automated evaluation detects at least 22% more errors than manual evaluation. The number of the detected errors is related to the number of elements in the SoS architecture. The more elements there are, the greater is the percentage difference between the errors detected by the automated means compared to the manual evaluation. Additionally, analysis of MBSE, UAF and automation by simulation means revealed their contribution to quality improvement by reducing the number of errors. The *Deloitte* survey disclosed that 92% of the respondents confirmed that automation and simulation met or exceeded their expectations, which resulted in improved quality/accuracy (90%).

SANTRAUKA

1 ĮVADAS

Motyvacija

Modeliais grindžiamos sistemų inžinerijos (angl. *Model-Based Systems Engineering*, MBSE) pritaikomumas ir toliau auga tiek privataus, tiek valstybinio kapitalo įmonėse; taip pat auga ir susidomėjimas MBSE kaip mokslinių tyrimų tema tarp sistemų inžinerijos (angl. *System Engineering*, SE) universitetų [2]. Maža to, MBSE išlieka pirmaujančia praktika tarptautinės SE tarybos (angl. *International Council on Systems Engineering*, INCOSE) 2035 m. vizijoje [3]. Viena iš pagrindinių MBSE plėtros priežasčių yra tai, jog MBSE leidžia kurti sistemas, turinčias atsekamumo ryšius su reikalavimais, taikant vieną integruotą sistemos architektūros modelį. Šie architektūros modeliai sudaro sąlygas tokių tipų automatizuotai analizei, kaip poveikio analizė (angl. *impact analysis*), alternatyvų palyginimas (angl. *trade study*) ir įvairios sistemos simuliacijos.

MBSE taikymas yra viena iš svarbiausių priemonių sprendžiant sudėtingas problemas [4], [5], [6]. Sistemos projektavimas tampa itin sudėtingas, kai nepriklausomos sistemos turi bendrauti tarpusavyje, kad pasiektų bendrą tikslą. Sistemų inžinieriai daugiausia dėmesio skiria sistemų projektavimui; vis dėlto dabar plačiai teigiama, jog sistemos ir prietaisai nebėra atskiri, bet tarpusavyje susiję kaip platesnė sistemų sistemos (angl. *System of Systems*, SoS) dalis. SE požiūriu tai vadinama SoS lygiu. Šiame lygmenyje viena iš pagrindinių problemų yra sistemos architektūros įvertinimas ir alternatyvų palyginimas.

Alternatyvų palyginimas yra nepakeičiama inžinerinė veikla, kuri dažnai praverčia ankstyvosiose sistemos kūrimo stadijose. Alternatyvų palyginimo metu atrinktos alternatyvios sistemų architektūros yra įvertinamos pasitelkiant atrankos kriterijus ir sisteminę analizę, siekiant nustatyti labiausiai subalansuotą alternatyvą iš kelių galimų [1], [7]. Alternatyvų palyginimo vykdymas kibernetinėms-fizinėms sistemoms (angl. *Cyber-physical systems*, CPS) ar SoS dažnai yra itin sudėtingas, jam prireikia daug laiko, išteklių ir žinių [8]. Ši inžinerinė veikla gali užtrukti nuo kelių mėnesių iki dvejų metų ir kainuoti nuo kelių šimtų tūkstančių iki keliolikos milijonų JAV dolerių; tai priklauso nuo palyginimo apimties ir sudėtingumo [9]. Neseniai atliktas JAV karinių oro pajėgų tyrimas parodė, jog vidutinės alternatyvų palyginimo sąnaudos siekia apie 15 milijonų JAV dolerių, o vidutinė trukmė – 21 mėnuo [9]. Tačiau, nepaisant didelių sąnaudų, alternatyvų palyginimas laikomas vertinga investicija, nes priimti netinkami sprendimai gali lemti žymiai išaugusias sąnaudas vėlesnėse sistemos kūrimo stadijose.

Įprastomis sąlygomis sistemų architektūros įvertinamos atskirai pagal kiekvieną dominantį kriterijų, o jų rezultatai lyginami rankiniu būdu [10]. Dažniausiai pasitaikantys trūkumai atliekant alternatyvų palyginimą yra skaidrumo trūkumas priimant galutinį sprendimą ir per didelis dėmesys mažiausios kainos kriterijui, kuris nebūtinai lemia geriausios vertės atitikimą [11]. Taip pat naujų alternatyvų pateikimas įpusėjus vertinimui sukelia nemažai iššūkių, nes tai žymiai padidina analizės sąnaudas lyginant su situacija, kai tokia pat alternatyvų apimtis būtų pasirinkta vertinimo pradžioje.

Šiuo metu aprašant SoS galima rinktis tarp keleto esamų sistemos architektūros aprašomųjų karkasų (angl. *Architecture Description Framework*, ADF), tokių kaip NATO architektūrinis karkasas (angl. *NATO Architecture Framework*, NAF) [12], [13] ar JAV gynybos ministerijos architektūrinis karkasas (angl. *Department of Defense Architecture Framework*, DoDAF) [14], [15], [16]. Laikui bėgant SoS inžinerija buvo plėtojama, siekiant supaprastinti sudėtingus sistemos kūrimo procesus. 2017 m. vieningas architektūrinis karkasas (angl. *Unified Architecture Framework*, UAF) tapo oficialiu OMG standartu [17]. UAF buvo atsižvelgiama į alternatyvų palyginimo veiklos svarbą ir sudarytos sąlygos alternatyvų palyginimui įvairiais lygmenimis: strateginiu (angl. *Strategic*, St), veiklos (angl. *Operational*, Op) ir išteklių (angl. *Resources*, Rs) [17]. Tačiau, nepaisant pateikto metamodelio, kuris skirtas surinkti ir specifikuoti reikiamus duomenis alternatyvų palyginimui, UAF neturi formalizuoto proceso, kuriuo būtų galima vadovautis atliekant alternatyvų palyginimą.

Literatūroje galima rasti keletą procesų, kurie aprašo alternatyvų palyginimą; tačiau nėra proceso, kuris aprašytų, kaip alternatyvų palyginimas galėtų būti atliktas MBSE aplinkoje kartu su esamais ADF, modeliavimo kalbomis ir architektūrų įvertinimo metodais. Sudėtingi modeliai, pažangi vizualizacija ir integruotos daugiadisciplinės simuliacijos leistų sistemų inžinieriams greičiau ir kruopščiau įvertinti daugiau alternatyvių sistemos architektūrų studijų.

Nors esami alternatyvų palyginimo metodai yra taikomi SE srityje, jų taikymas SoS architektūroms MBSE aplinkoje nebuvo išsamiai ištirtas. Be to, spartėjant skaitmeninimui ir didėjant MBSE naudojančių organizacijų skaičiui, trūksta moksliškai įrodytų alternatyvų palyginimo metodų, kuriuos būtų galima taikyti MBSE aplinkoje [18], [19].

Disertacijos objektas ir sritis

Šios disertacijos objektas yra alternatyvų palyginimo metodas, skirtas įvertinti SoS sprendimo architektūras MBSE aplinkoje.

Tyrimo sritis:

- sistemų sistemos inžinerija (angl. *System of Systems Engineering*, SoSE) ir MBSE;
- SoS modeliavimo metodai ir karkasai;
- alternatyvų įvertinimo metodai;
- standartai ir metodai, skirti vykdyti automatizuotą architektūrų įvertinimą MBSE aplinkoje.

Disertacijos tikslas

Pagerinti alternatyvių SoS architektūrų palyginimo kokybę pateikiant priemones, leidžiančias automatizuoti SoS architektūrų įvertinimą pasitelkiant sistemos simuliaciją MBSE aplinkoje.

Disertacijos uždaviniai

Siekiant įgyvendinti iškeltą tikslą, buvo suformuluoti penki uždaviniai.

1. Išanalizuoti esamus architektūrų aprašomuosius karkasus, sutelkiant dėmesį į jų pritaikymą alternatyvų palyginimui ir į galimybes vykdyti automatizuotą SoS architektūrų įvertinimą.
2. Išanalizuoti esamus alternatyvų palyginimo procesus ir architektūrų įvertinimo metodus, sutelkiant dėmesį į jų praktinį taikymą MBSE aplinkoje.
3. Parengti modeliais grįstą alternatyvų palyginimo metodą, kuris apimtų alternatyvių SoS sprendimo architektūrų įvertinimą MBSE aplinkoje.
4. Įgyvendinti parengtą metodą CASE įrankiu.
5. Eksperimentiškai įvertinti parengto metodo tinkamumą ir praktinį pritaikomumą.

Tyrimo metodika

Šios disertacijos tyrimo metodika yra pagrįsta tradiciniu mokslinio tyrimo procesu [20], kurį galima suskirstyti į tris pagrindines dalis.

Pirmoje dalyje analizuojama esama literatūra apie SE, MBSE, ADF, alternatyvų palyginimo procesus ir alternatyvių sistemos architektūrų įvertinimo metodus. Peržiūrėta literatūra sistemingai apibendrinta taikant palyginimo ir klasifikavimo metodus.

Antroje dalyje, kuriant alternatyvių SoS sprendimo architektūrų palyginimo metodą, kuris taikytinas MBSE aplinkoje, taikomi šie metodai: konceptualūs modeliavimo metodai, UAF domeno metamodelis, sintezė, integracija ir metamodeliavimas.

Trečioje dalyje atvejo tyrimas (angl. *case study*) atliekamas vertinant šioje disertacijoje parengtą metodą. Atvejo tyrimas sumodeliuotas pagal parengto metodo pateiktas gaires ir eksperimentiškai įvertintas atliekant alternatyvų palyginimą MBSE aplinkoje. Eksperimentinis vertinimas apima kiekybinę duomenų analizę ir klasifikavimo metodus.

Ginamieji teiginiai

Disertacijos ginamieji teiginiai yra trys.

1. MBSE aplinka ir pamatinis UML poaibis (angl. *foundational UML subset*, fUML) yra tinkamos priemonės automatizuoti alternatyvių SoS sprendimo architektūrų įvertinimą siekiant sumažinti architektūrų įvertinimo klaidas.
2. Esamas UAF metamodelis turi būti išplėstas naujomis koncepcijomis, siekiant apibrėžti ir įvykdyti alternatyvų palyginimą MBSE aplinkoje.
3. Suformuluotas UAF grįstas alternatyvių SoS architektūrų palyginimo metodas (UT3SA) pateikia reikiamą procesą, gaires, įvertinimo algoritmus ir validacijos taisykles, kurios leidžia pagerinti alternatyvų palyginimo kokybę sumažinant klaidų kiekį, kai vykdomas SoS architektūrų įvertinimas.

Mokslinis naujumas

Šios disertacijos mokslinį naujumą galima apibrėžti keliais punktais.

1. UT3SA yra vienintelis žinomas metodas, apimantis visas šias savybes.
   - 1.1. Pasiūlytas alternatyvų palyginimo procesas yra pagrįstas oficialiu UAF metamodeliu ir profiliu. Taip pat procesas apima būtinus žingsnius, kai alternatyvų palyginimas atliekamas kaip pirkimų proceso rezultatas.
   - 1.2. Vertikalus (orientuotas į įvairias sritis) ir horizontalus (orientuotas į vieną sritį) alternatyvių SoS sprendimo architektūrų kokybės įvertinimas.
   - 1.3. Kokybinių ir kiekybinių SoS architektūrų parametrų įvertinimas.
   - 1.4. Šis metodas grįstas modeliais, ir tai sudaro sąlygas jo taikymui MBSE aplinkoje.
   - 1.5. Suformuluotas metodas apibrėžia du algoritmus, kurie leidžia atlikti automatizuotą SoS sprendimo architektūrų įvertinimą ir jautrumo analizę MBSE aplinkoje. Algoritmai yra pagrįsti fUML principais.
2. Esamas UAF metamodelis ir profilis yra papildyti būtinais plėtiniais, kad būtų galima parengti ir vykdyti alternatyvių SoS sprendimo architektūrų palyginimą MBSE aplinkoje.
3. Suformuluotas alternatyvų palyginimo metodas yra vienas iš pirmųjų metodų, skirtų SoS sprendimo architektūroms palyginti MBSE aplinkoje, paskelbimo metu.

Praktinė reikšmė

Pagrindinė šio tyrimo praktinė reikšmė – pagerinta alternatyvių SoS sprendimo architektūrų palyginimo kokybė, naudojant MBSE aplinką. Suformuluotas metodas pagerino alternatyvų palyginimo kokybę suteikdamas šias priemones.

- Aprašyti ir susieti alternatyvų palyginimo reikalavimus su numatoma SoS sprendimo architektūra. Sukurtas metodas pateikia gaires, kaip apibrėžti alternatyvų palyginimą MBSE aplinkoje naudojantis UAF metamodeliu. Alternatyvų palyginimo reikalavimų susiejimas su numatoma SoS sprendimo architektūra MBSE aplinkoje pagerina alternatyvų palyginimo tikslumą.
- Nustatyti ir atmesti nepakankamos kokybės SoS sprendimo architektūras prieš atliekant išsamų kiekvienos alternatyvios SoS sprendimo architektūros įvertinimą. Alternatyvių SoS sprendimo architektūrų kokybės įvertinimas pagerina analizės kokybę, nes galutiniai palyginimo rezultatai gali būti iškreipti dėl netikslios sistemos architektūros kokybės.
- Įvertinti alternatyvių SoS sprendimo architektūrų atitikimo lygmenį pagal nustatytus atrankos kriterijus MBSE aplinkoje. Automatizuotas įvertinimo metodas sumažina klaidų atsiradimo tikimybę bei sudaro sąlygas atlikti paprastesnį alternatyvų palyginimo apimties keitimą nei taikant tradicinius alternatyvų palyginimo būdus.

- Identifikuoti alternatyvias SoS architektūras, kurios turi būti patikrintos atliekant jautrumo analizę. Jautrumo analizė atliekama siekiant nustatyti, kuri alternatyva turi didesnį pranašumą, kai atitikties lygis yra labai panašus. Jautrumo analizės įtraukimas į alternatyvų palyginimo procesą pagerina tyrimo kokybę, įvedant papildomą įvertinimo etapą, kuris skirtas verifikuoti galutiniam sistemos alternatyvos pasirinkimo sprendimui.
- Nustatyti kiekvieno atrankos kriterijaus jautrumo indeksą. Labiausiai jautrūs kriterijai yra atrenkami tolimesnei analizei, siekiant palyginti alternatyvias SoS sprendimo architektūras, kurios turi panašų įvertį. Šis įvertinimas pagerina labiausiai kriterijus atitinkančios alternatyvos pasirinkimo tikslumą.

Metodo taikymas aštuonioms alternatyvioms SoS sprendimo architektūroms parodė, jog sukurtas metodas gali būti taikomas praktikoje. Norint pritaikyti sukurtą metodą, reikia jį įdiegti viename iš SoS modeliavimo CASE įrankių. Šios disertacijos metu buvo sukurtas UAF grįstas UT3SA profilis, kuris įgyvendintas naudojant *Catia Magic Systems of Systems Architect v2021x Refresh2* (MSoSA) CASE įrankį.

Sukurtas metodas yra grįstas UAF principais. Kadangi UAF yra UML/SysML modeliavimo kalbų plėtinys, galima daryti prielaidą, jog sukurtas metodas gali būti adaptuotas ir SE lygmeniu.

Rezultatų aprobavimas

Disertacijos mokslinių tyrimų rezultatai buvo pristatyti dviejuose straipsniuose, kurie išspausdinti recenzuojamuose mokslo leidiniuose, turinčiuose cituojamumo rodiklį duomenų bazėje *Clarivate Analytics Web of Science* (CA WoS). Taip pat disertacijos rezultatai buvo pristatyti keturiose tarptautinėse konferencijose, kurios vyko Vengrijoje, Pietų Afrikos Respublikoje (PAR), JAV ir Lietuvoje surengtame tarptautiniame seminare. Išsamus publikacijų sąrašas pateiktas 7 skyriuje „Autorės publikacijų disertacijos tema sąrašas“.

Disertacijos struktūra

Disertacijos pirmame skyriuje pateikiama SoSE, MBSE, ADF, alternatyvų palyginimo procesų ir architektūrų įvertinimo metodų analizė. Antrame skyriuje apibrėžiamas siūlomas UT3SA metodas, skirtas SoS sprendimo architektūroms palyginti MBSE aplinkoje. Trečiame skyriuje pateikiamas sukurto UT3SA metodo tinkamumo įvertinimas, taikant jį aštuonių alternatyvių RV sprendimo architektūrų palyginimui MBSE aplinkoje.

2 MOKSLINĖS LITERATŪROS ANALIZĖ

Šis skyrius apima mokslinės literatūros analizės apžvalgą, kuri buvo atlikta tiriant SoSE, MBSE, ADF, alternatyvų palyginimo procesus ir jiems taikomus metodus.

2.1 Sistemų sistemos inžinerija ir modeliais grindžiama sistemų inžinerija

Tradicinė SE ir SoSE yra tarpusavyje susijusios inžinerijos sferos, tačiau jos apima skirtingas veiklos sritis. SE apibrėžiama kaip „procesas, kurio metu patenkinami kliento poreikiai konceptualizuojant, projektuojant, modeliuojant, testuojant, įgyvendinant ir eksploatuojant sistemą“ [21]. O štai SoSE apibrėžiama kaip „esamų ir naujų sistemų derinio planavimo, analizės, organizavimo ir integravimo į SoS pajėgumą procesas“ [21].

Iš pradžių SoSE buvo taikoma tik gynybos sektoriui (angl. *Department of Defense*, DoD), visgi šiandien SoSE koncepcijos bei principai taikomi ir valstybinėse bei komercinėse organizacijose. SE žinių rinkinyje (SEBoK) [22] teigiama, jog SoSE suteikia SE bendruomenei galimybę kurti sudėtingas sistemas. Vienas esminių SE ir SoSE skirtumų yra tai, jog SE apima monolitinių sistemų kūrimą, o SoSE – sudėtinių sistemų kūrimą, kur šių sistemų sinergija vadinama SoS. Kitaip tariant, SoS sudaro nepriklausomos sistemos, kurios veikia kartu siekdamos bendro tikslo [31].

Ilgą laiką, apie keturiasdešimt metų, SE praktika buvo linijinė: 1) reikalavimų dokumentacija; 2) analizės vykdymas; 3) koncepcinio projekto kūrimas. Sistemos kūrimo ciklo (angl. *life cycle*) pažanga įvedė naujus metodus projektuojant ir tobulinant sistemas, tokias kaip krioklys, spiralė, prieaugis, bei pagrįstas sprintais. Tačiau sistemos integracijų problemos tarp skirtingų kūrimo etapų lėmė itin pailgėjusį kūrimo laiką.

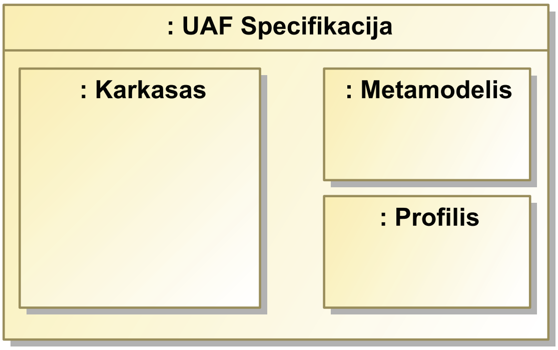
„MBSE yra formalizuotas modeliavimo taikymas, kuris palaiko sistemos reikalavimus, projektavimo, analizės, reikalavimų verifikacijos ir validavimo veiklas, pradedant nuo konceptualaus projektavimo etapo ir tęsiant per visą sistemos kūrimo etapų laikotarpį“ [3].

OMG organizacija išskiria penkis MBSE privalumus, kuriuos suteikia SE [60]:

- pagerina tikslumą, nuoseklumą ir atsekamumą;
- apima elgesio analizę, sistemos architektūrą, reikalavimų atsekamumą, našumo analizę, simuliaciją, testavimą;
- formalizuoja sistemos kūrimo praktikas taikant modelius;
- integruoja duomenis tarp specifinių inžinerinių įrankių;
- supaprastina bendrą sistemos supratimą.

2.2 Vieningas architektūros karkasas

Atsižvelgiant į pramonės poreikius ir ADF raidą, 2017 m. buvo pristatytas naujas karkasas – UAF, kuris tapo oficialiu OMG standartu [17]. UAF pagrindiniai konceptai yra pagrįsti trimis gynybiniais ADFS: DoDAF, MODAF ir NAF. Tačiau vieningo DoDAF/MODAF profilio (angl. *Unified Profile for DoDAF/MODAF*, UPDM) praplėtimas ir jo apibendrinimas UAF leidžia taikyti tiek komercinei, tiek gynybos sistemų architektūrai“ [46]. UAF architektūros modeliai suteikia galimybę suprasti sudėtingus santykius tarp organizacijų, sistemų ir SoS bei sudaro sąlygas vykdyti įvairaus pjūvio sistemų analizes [18]. UAF kaip karkasas ir UAFP kaip kalba buvo taikoma įvairių SoS atvejų kūrimui.

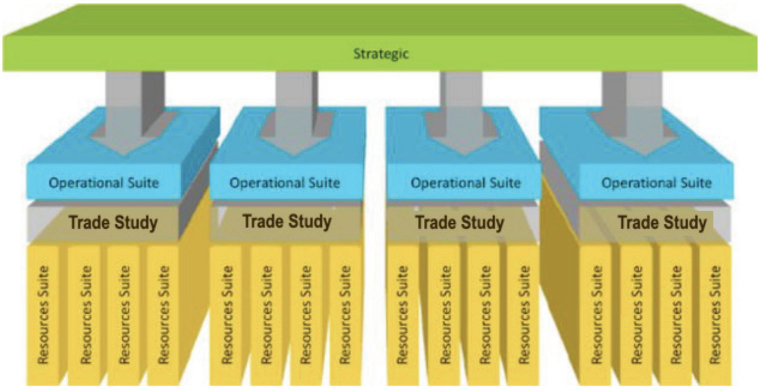


1 pav. UAF komponentai

UAF susideda iš trijų pagrindinių komponentų [46], [18] (1 pav.):

- **karkaso** – sričių, modelių tipų ir aspektų (angl. *viewpoints*) grupės;
- **metamodelio** – tipų grupės, skirtos vaizdiniams (angl. *views*) kurti remiantis konkrečiais sistemos pjūviais;
- **profilio** (UAFP) – UML 2.5.1 ir SysML 1.5 grįsto metamodelio įgyvendinimo, siekiant taikyti MBSE principus ir geriausią praktiką kuriant vaizdus.

Trys UAF sritys yra laikomos pagrindinėmis, atspindinčiomis skirtingus abstrakcijos lygius, ir šios sritys yra St, Op ir Rs [55]. Paprastai architektūros kūrimas vyksta nuo St srities iki Rs srities. Kuriant sistemos architektūrą, alternatyvų palyginimas gali būti vykdomas tarp skirtingų Op scenarijų ir / ar skirtingų Rs konfigūracijų (2 pav.).



2 pav. ADF pagrindiniai abstrakcijos lygiai [57]

Tačiau UAF neapibrėžia, kaip atliekamas alternatyvų palyginimas, ir nepateikia alternatyvų palyginimo proceso. Tik suteikia galimybę specifikuoti duomenis, reikalingus alternatyvoms palyginti, ir sudaro sąlygas fUML konceptus panaudoti automatizuotai analizei vykdyti MBSE aplinkoje [18], [80], [81].

2.3 Alternatyvų palyginimo procesai ir metodai

Alternatyvų palyginimas (angl. *Trade Study*) yra įprasta SE užduotis, kuri apibrėžiama taip: „SE dalis, kurią daugiadisciplininės komandos naudoja siekdamos nustatyti labiausiai subalansuotiems techniniams sprendimams iš siūlomų alternatyvių sprendimų rinkinio. Tai pagrindinė analizės priemonė įvairiuose civiliniuose ir gynybos projektuose norint atitikti suinteresuotųjų šalių sistemos keliamus reikalavimus ekonomiškiausiu būdu“ [83]. Alternatyvų palyginimas yra formali priemonė, padedanti priimti sprendimus ir nustatyti, ar alternatyvos atitinka pageidaujamo sprendimo kriterijus, ar ne.

Pasaulinė SE bendruomenė, vykdydama alternatyvų palyginimą, taiko įvairius alternatyvų palyginimo procesus, kurie buvo pasiūlyti NASA [85], NAF [40], Reiter [86], MITRE [87]. Tačiau esami alternatyvų palyginimo procesai nėra skirti konkrečiai sričiai ar pramonės šakai, tad juose pateikiami bendri žingsniai ir vartojama bendra terminologija. Taip pat esami procesai apibrėžia tik pagrindinę alternatyvų palyginimo eigą, neįtraukdami alternatyvų įvertinimo detalių, tokių kaip konkretūs įvertinimo metodai, technikos, algoritmai. Ir kartu esamų alternatyvų palyginimo procesų taikymas MBSE aplinkoje nėra formalizuotas.

Alternatyvų palyginimas dažniausiai apima dvi analizes: daugiakriterių sprendimų analizė (angl. *Multi-criteria Decision Analysis*, MCDA) ir jautrumo analizė. MCDA yra priemonė, padedanti priimti sprendimus, kai susiduriama su didelio kriterijų kiekio vertinimu [88]. MCDA tikslas – išryškinti problemas tarp įvairių vertinimų ir rasti būdą, kaip pasiekti kompromisus [89]. Penki plačiausiai taikomi MCDA metodai palyginti 1 lentelėje, apibendrinant jų pagrindinius privalumus ir trūkumus.

1 lentelė. MCDM metodų apžvalga

Metodas	Privalumai	Trūkumai
MAUT	▪ Atsižvelgiama į neapibrėžtumą ▪ Įtraukiami prioritetai	▪ Reikalingas didelis duomenų rinkinys ▪ Prioritetai turi būti tikslūs
Pugh	▪ Paprastas ▪ Nereikalingas didelis duomenų rinkinys	▪ Reikšmių priskyrimas yra subjektyvus ▪ Kriterijų sąrašas yra sutartinis
MVA	▪ Nagrinėjami skirtingi nepriklausomieji kintamieji, turintys įtakos priklausomiesiems kintamiesiems ▪ Lengva taikyti dirbant su statistinių duomenų paketais	▪ Būtini sudėtingi matematiniai skaičiavimai ▪ Reikalingas didelis duomenų rinkinys
AHP	▪ Paprastumas ▪ Geresnis išplečiamumas (angl. *Scalability*) ▪ Hierarchijos struktūrą galima lengvai pritaikyti prie daugelio problemų ▪ Nereikia daug duomenų	▪ Kriterijų ir alternatyvų tarpusavio priklausomybė gali daryti įtaką sprendimo ir reitingavimo kriterijų neatitikimui ▪ Ekspertai nustato kriterijų svorius

Metodas	Privalumai	Trūkumai
TOPSIS	- Paprastumas - Skaičiavimo efektyvumas - Naudojamos matematinės formos alternatyvoms lyginti	- Taikant Euklido atstumą neatsižvelgiama į atributų koreliaciją - Sunku įvertinti ir išlaikyti sprendimo nuoseklumą

1 lentelėje parodyta, kad Pugh, AHP ir TOPSIS metodus lengva taikyti, o štai dirbant MAUT ir MVA metodais reikalingi dideli duomenų rinkiniai ir sudėtingi matematiniai skaičiavimai. AHP yra vienintelis metodas, kurį galima taikyti esant dideliam kiekiui kriterijų dėl jo išplečiamumo ir struktūros.

Alternatyvų palyginimo rezultatai gali parodyti keletą alternatyvų su panašiais ar net lygiaverčiais balais. Tokiu atveju reikėtų atlikti jautrumo analizę, siekiant įvertinti alternatyvos pasirinkimą. Jautrumo analizė yra metodas, tiriantis įvesties neapibrėžtumo poveikį atsakui į modelio išvestį [116]. Jautrumo analizė taikoma sistemos rezultatų patikimumui patikrinti, taip pat norint suprasti ryšį tarp kintamųjų ir jų santykinę įtaką sistemos veikimui [118]. Jei keičiant įvesties kintamąjį yra didelių išvesties rezultato skirtumų, tada teigiama, kad išvestis yra jautri. Jei išvestis reikšmingai nesikeičia, išvestis yra nejautri.

Jautrumo analizės metodai skirstomi į dvi pagrindines grupes: vietinę ir globalią. Vietinė jautrumo analizė yra modelio išvesties pokyčių vertinimo metodas, grįstas vieno parametro įvesties pokyčiais [123]. Keičiamas tik vienas įvesties parametras (angl. *One-At-a-Time*, OAT), o kiti parametrai paliekami nepakitę. Globali jautrumo analizė tuo pačiu metu keičia visus parametrus. Tai leidžia vienu metu įvertinti santykinį kiekvieno parametro indėlį ir parametrų sąveiką su modelio išvesties nuokrypiu [122]. Vietinė jautrumo analizė tinka, kai norima nustatyti konkretaus atrankos kriterijaus jautrumą. O štai atliekant globalią jautrumo analizę vienu metu yra keičiami keli atrankos kriterijai, todėl negalima sužinoti apie konkretaus atrankos kriterijaus jautrumą.

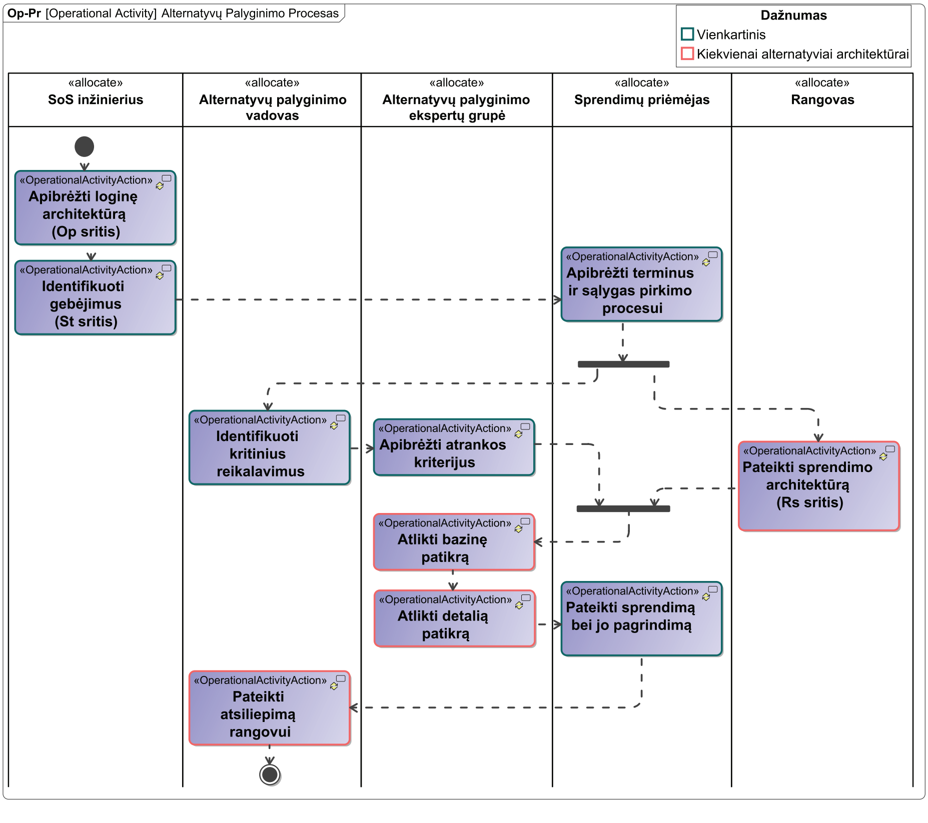
3 UT3SA: ALTERNATYVIŲ SISTEMŲ ARCHITEKTŪRŲ PALYGINIMO METODAS, GRĮSTAS VIENINGU ARCHITEKTŪROS KARKASU, MODELIAIS GRINDŽIAMOJE SISTEMŲ SISTEMOS INŽINERIJOJE

UT3SA metodas formalizuotas pasirinkus du OMG standartus: UAF v1.2 ir fUML v1.5. UT3SA alternatyvų palyginimo profilis ir susiję plėtiniai (lentelės, algoritmai, validacijos taisyklės ir metrikos) parengti su *Catia MSoSA v2021xRefresh2* įrankiu, kurie gali būti įdiegti kaip įskiepis į bet kurį suderinamą SoS CASE įrankį.

Siūlomu UT3SA metodu siekiama pagerinti alternatyvių SoS sprendimo architektūrų palyginimo kokybę, suteikiant priemones įvertinti architektūras MBSE aplinkoje. Alternatyvų palyginimo kokybės gerinimas reiškia galimų klaidų mažinimą vertinant alternatyvias SoS architektūras. Šis metodas padeda nustatyti labiausiai sistemos reikalavimus atitinkančią sprendimų architektūrą iš kelių galimų alternatyvų. Siekiant automatizuoti analizės vykdymą MBSE aplinkoje, pasirinkta simuliacijos technika, kuri yra grįsta fUML standartu. Taip pat UT3SA metodas pateikia gaires, kaip parengti simuliuojamą modelį automatizuotam alternatyvų palyginimui MBSE aplinkoje. UT3SA metodas susideda iš dviejų pagrindinių dalių: 1) UAF grįstas alternatyvų palyginimo procesas; 2) alternatyvų palyginimo analizės algoritmai, validacijos taisyklės ir UT3SA metodo taikymo gairės MBSE aplinkoje.

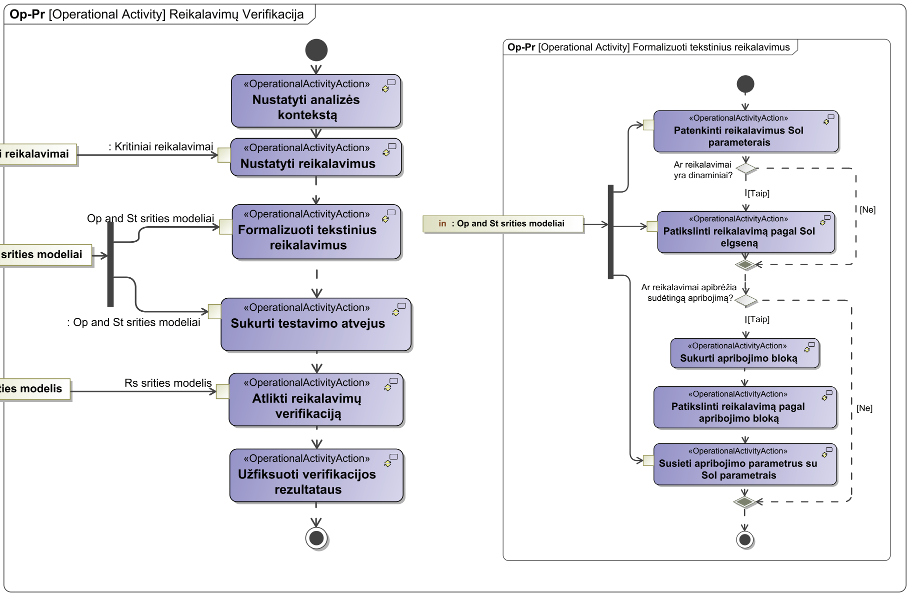
Pasiūlytas alternatyvų palyginimo procesas parengtas pagal UAF principus, jis sujungia automatizuotus analizės etapus. Taip pat šis procesas apima reikiamus žingsnius, kai alternatyvų palyginimas atliekamas kaip pirkimų proceso rezultatas. Alternatyvų palyginimo procesas yra išskaidytas į dešimt pagrindinių žingsnių (3 pav.). Kiekvienas proceso žingsnis išsamiai aprašytas pateikiant vaidmenį, kuris atlieka reikalaujamą žingsnį, apibrėžiant įvesties / išvesties duomenis ir nustatant proceso eiliškumo tvarką. UAF grįstas alternatyvų palyginimo procesas pristatytas straipsnyje, išspausdintame konferencijos medžiagoje *INCOSE International Symposium* [57].

Bazinė patikra (angl. *Base Check*) yra viena iš dviejų analizių, vykdomų alternatyvų palyginimo procese (3 pav.); ji atliekama pasitelkiant sistemos simuliaciją. Šio analizės etapo tikslas – patikrinti, ar SoS sprendimo architektūros atitinka kritinius reikalavimus, ir nustatyti SoS architektūros kokybę (SoS architektūros modelio išsamumo (angl. *completeness*) ir teisingumo (angl. *correctness*) patikra). Bazinės patikros etapas susideda iš dviejų dalių: 1) kritinių reikalavimų atitikimo patikra; 2) kokybės įvertinimas.


Op-Pr [Operational Activity] Alternatyvų Palyginimo Procesas
Dažnumas
□ Vienkartinis
□ Kiekvienai alternatyviai architektūrai
«allocate» SoS inžinierius
«allocate» Alternatyvų palyginimo vadovas
«allocate» Alternatyvų palyginimo ekspertų grupė
«allocate» Sprendimų priėmėjas
«allocate» Rangovas
«OperationalActivityAction» Apibrėžti loginę architektūrą (Op sritis)
«OperationalActivityAction» Identifikuoti gebėjimus (St sritis)
«OperationalActivityAction» Apibrėžti terminus ir sąlygas pirkimo procesui
«OperationalActivityAction» Identifikuoti kritinius reikalavimus
«OperationalActivityAction» Apibrėžti atrankos kriterijus
«OperationalActivityAction» Pateikti sprendimo architektūrą (Rs sritis)
«OperationalActivityAction» Atlikti bazinę patikrą
«OperationalActivityAction» Atlikti detalią patikrą
«OperationalActivityAction» Pateikti sprendimą bei jo pagrindimą
«OperationalActivityAction» Pateikti atsiliepimą rangovui

3 pav. UAF grįstas alternatyvų palyginimo procesas

Kritinių reikalavimų patikra vykdomas analizuojant atsekamumo ryšius pagal parengtas taisykles tarp dviejų abstrakcijos lygių: loginio ir resursų. Tačiau automatizuotam kiekybinių reikalavimų tikrinimui yra parengtas naujas reikalavimų verifikacijos procesas, kuris apima tiek parametrinių, tiek dinaminių reikšmių tikrinimą (4 pav.). Kritinių reikalavimų patikros procesas pristatytas straipsnyje, išspausdintame konferencijos medžiagoje *INCOSE International Symposium* [134].

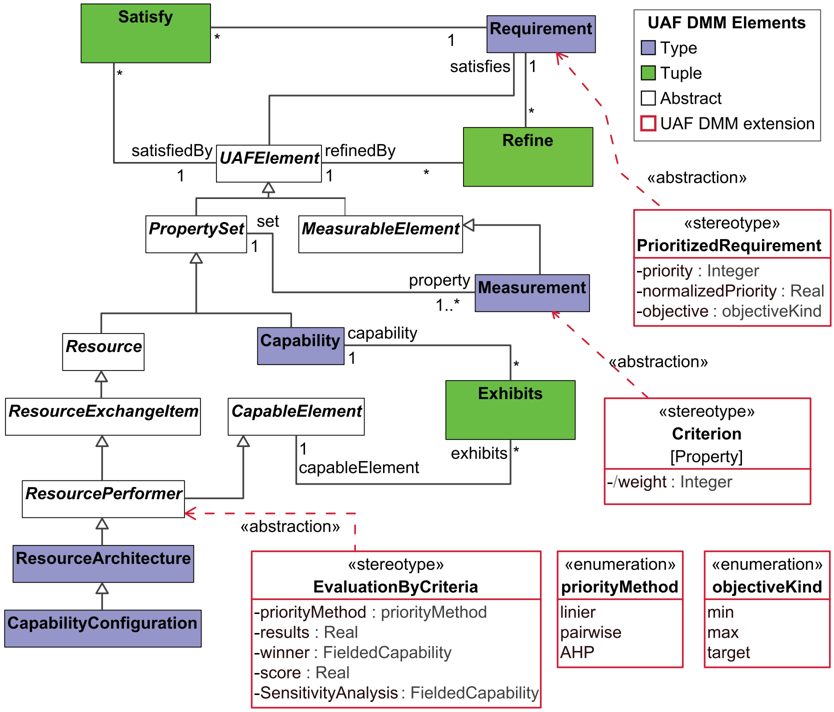

Op-Pr [Operational Activity] Reikalavimų Verifikacija
«OperationalActivityAction»
Nustatyti analizės kontekstą
in : Kritiniai reikalavimai
: Kritiniai reikalavimai
«OperationalActivityAction»
Nustatyti reikalavimus
Op and St srities modeliai
«OperationalActivityAction»
Formalizuoti tekstinius reikalavimus
in : Op and St srities modeliai
: Op and St srities modeliai
«OperationalActivityAction»
Sukurti testavimo atvejus
in : Rs srities modelis
Rs srities modelis
«OperationalActivityAction»
Atlikti reikalavimų verifikaciją
«OperationalActivityAction»
Užfiksuoti verifikacijos rezultataus
Op-Pr [Operational Activity] Formalizuoti tekstinius reikalavimus
in : Op and St srities modeliai
«OperationalActivityAction»
Patenkinti reikalavimus Sol parameterais
Ar reikalavimai yra dinaminiai?
[Taip]
[Ne]
«OperationalActivityAction»
Patikslinti reikalavimą pagal Sol elgseną
Ar reikalavimai apibrėžia sudėtingą apribojimą?
[Taip]
[Ne]
«OperationalActivityAction»
Sukurti apribojimo bloką
«OperationalActivityAction»
Patikslinti reikalavimą pagal apribojimo bloką
«OperationalActivityAction»
Susieti apribojimo parametrus su Sol parametrais

4 pav. Kiekybinių reikalavimų verifikacijos procesas

Alternatyvių SoS architektūrų kokybės įvertinimas vykdomas analizuojant atsekamumo ryšius tarp trijų UAF sričių: St, Op ir Rs. Automatizuotam kokybės įvertinimui buvo parengti du validacijos taisyklių rinkiniai: 1) vertikalusis architektūrų kokybės įvertinimas; 2) horizontalusis architektūrų kokybės įvertinimas. Vertikaliojo kokybės įvertinimo taisyklės skirtos patikrinti alternatyvioms SoS sprendimo architektūroms pagal sistemos reikalavimus. Šiuo įvertinimu tikrinami St, Op ir Rs sričių modeliai bei jų atsekamumo ryšiai. Taisyklių rinkinys suskirstytas į dvi kategorijas pagal tikrinamas sritis (St ir Rs; Op ir Rs). O štai horizontaliojo kokybės įvertinimo taisyklės skirtos izoliuotai patikrinti alternatyvioms SoS sprendimo architektūroms, analizuojant tik Rs modelius ir jų atsekamumo ryšius. Kiekviena taisyklė turi nustatytą tikrinamąjį elementą, svarbumo lygį ir jo pagrindimą. Svarbumui žymėti pasirinkti trys lygiai: aukštas (svoris = 3), vidutinis (svoris = 2) ir žemas (svoris = 1). SoS sprendimo architektūrų kokybės įvertinimo principai ir jų eksperimentinis taikymas buvo pristatyti straipsnyje, išspausdintame mokslo žurnale *IEEE Access* [56].

Detalioji patikra (angl. *Deep Check*) yra antrasis analizės etapas, kuris vykdomas alternatyvų palyginimo procese (3 pav.) pasitelkiant sistemos simuliaciją. Šio analizės etapo tikslas – įvertinti atrinktas alternatyvias SoS sprendimo architektūras pagal atrankos kriterijus ir išrinkti labiausiai atitinkančią kriterijus architektūrą. Detalioji patikra apima tris dalis: 1) pradinių duomenų specifikavimą MBSE aplinkoje; 2) alternatyvių architektūrų įvertinimą pagal atrankos kriterijus; 3) jautrumo analizės vykdymą. Detaliosios patikros principai ir jų eksperimentinis taikymas buvo pristatyti straipsnyje, išspausdintame mokslo žurnale *IEEE Access* [131].

Siekiant vykdyti sistemos architektūrų įvertinimą MBSE aplinkoje, atrankos kriterijai, jų svarbumo lygiai ir alternatyvios SoS sprendimo architektūros turi būti specifikuotos modelyje. Kadangi UAFP neapibrėžia reikiamų konceptų atrankos kriterijams specifikuoti, tam buvo pasiūlyti UAFP plėtiniai, kurie užtikrina reikiamų duomenų specifikavimą modelyje, siekiant vykdyti automatizuotą architektūrų įvertinimą (5 pav.).



5 pav. UAF DMM plėtiniai atrankos kriterijų specifikavimui MBSE aplinkoje

Alternatyvių SoS sprendimo architektūrų įvertinimui pagal atrankos kriterijus MBSE aplinkoje buvo parengtas algoritmas, grįstas fUML principais. Vykdant algoritmą, kiekviena alternatyvi architektūra yra įvertinama pagal nustatytus atrankos kriterijus ir apskaičiuojamas galutinis bendras įvertis. Tačiau kelių sistemos architektūrų galutinis įvertis gali būti labai panašus ar net vienodas. Siekiant užtikrinti labiausiai kriterijus atitinkančios architektūros pasirinkimą, tokiu atveju rekomenduojama atlikti jautrumo analizę. Alternatyvos, kurios gavo iki 5 % mažesnį balą nei aukščiausią balą gavusi alternatyva, atrenkamos jautrumo analizei, įskaitant patį aukščiausią balą gavusią alternatyvą [138], [139], [140], [141]. Automatizuotam jautrumo analizės vykdymui MBSE aplinkoje buvo parengtas OAT jautrumo analizės algoritmas, grįstas fUML principais. Vykdant algoritmą, alternatyvių sistemos architektūrų, kurios gavo panašų galutinį įvertį (<5 %), atrankos kriterijų atitikimo balas yra modifikuojamas pagal apibrėžtą jautrumo kintamąjį. Jautrumo analizės metu pagal analizuojamą architektūrą apskaičiuojami atrankos kriterijų jautrumo indeksai. Galiausiai atliktų analizių rezultatai yra peržiūrimi, siekiant atrinkti sistemos architektūrą, kuri geriausiai atitinka iškeltus reikalavimus.

Kadangi literatūroje nėra publikuoto panašios charakteristikos alternatyvų palyginimo metodo kaip UT3SA, šio metodo palyginimas buvo atliktas fragmentiškai lyginant dvi pagrindines jo dalis: procesą ir SoS architektūrų įvertinimo metodus.

Pasiūlytas UAF grįstas alternatyvų palyginimo procesas buvo palygintas su kitais alternatyvų palyginimo procesais: NASA, NAF, Reiter ir MITRE.

2 lentelė. Alternatyvų palyginimo procesų palyginimas

Kriterijai		NASA	NAF	Reiter	MITRE	UT3SA
Aprašo vaidmenis		–	–	–	–	+
Aprašo įvesties / išvesties duomenis		–	–	–	–	+
Grįstas formaliu metamodeliu		–	+	–	–	+
Pagrindiniai alternatyvų palyginimo proceso etapai	**Pasiruošimas**	+	+	+	+	+
	Kritinių kriterijų specifikavimas	–	–	–	–	+
	Atrankos kriterijų specifikavimas	+	+	+	+	+
	Įvertinimas	+	+	+	+	+
	Etapas dėl įsigijimo proceso vykdymo	–	–	–	–	+
	Išvados	–	+	+	+	+
Įvertinimo etapo analizės metodai (Ds – aprašomasis, Dt – detalus)		Ds	Ds	Ds	Ds	Dt
MBSE palaikymas		–	–	–	–	+

2 lentelėje pateikti rezultatai atskleidė, jog UAF grįstas UT3SA alternatyvų palyginimo procesas yra vienintelis procesas, kuris apibrėžia kiekvieno etapo įvesties / išvesties duomenis ir vaidmenį, taip pat apima reikiamus žingsnius, kai alternatyvų palyginimas yra atliekamas kaip pirkimų proceso rezultatas. UT3SA procese pateikiami išsamūs alternatyvių sistemos architektūrų įvertinimo metodai, kurie apima algoritmus, validacijos taisykles ir gaires, kaip apibrėžti alternatyvų palyginimą MBSE aplinkoje. Dar vienas pranašumas, lyginant su kitais procesais, yra tas, jog UT3SA procesas grįstas formaliu metamodeliu.

UT3SA apima tris sistemos architektūrų įvertinimus: 1) kokybės įvertinimą, grįstą validacijos taisyklėmis; 2) įvertinimą pagal atrankos kriterijus, grįstą AHP metodu; 3) jautrumo analizę, grįstą OAT metodu. UT3SA taikomi architektūros įvertinimo metodai buvo palyginti su grindžiamais metodais (MCDA: AHP ir SA: OAT).

3 lentelė. Architektūrų įvertinimo metodų palyginimas

Kriterijai	MCDA: AHP + SA: OAT	UT3SA
Kokybinių ir kiekybinių SoS architektūros parametrų analizė	+	+
SoS architektūros modelio teisingumo ir išsamumo analizė	–	+
Kiekybinis įvertis	+	+
MBSE palaikymas	–	+ (Įtraukti algoritmai)

3 lentelėje pateikti rezultatai rodo, jog UT3SA siūlomi architektūros įvertinimo metodai yra pranašesni už AHP ir OAT metodų derinį. UT3SA įvertinimo metodai suteikia du privalumus: 1) SoS architektūros modelio teisingumo ir išsamumo įvertinimą; 2) MBSE palaikymą.

4 EKSPERIMENTINIS ĮVERTINIMAS

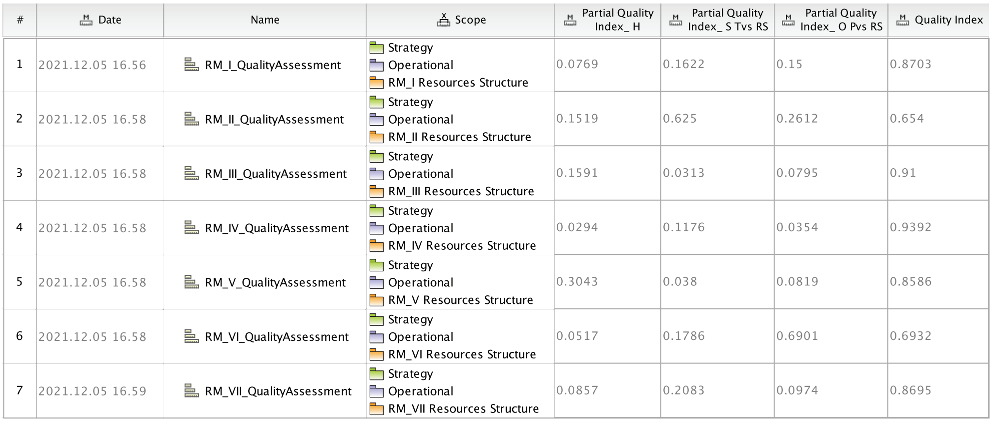
Sukurtas UT3SA metodas eksperimentiškai įvertintas pasitelkiant atvejo tyrimą, kuris buvo vykdomas *Catia MSoSA* CASE įrankiu. Sukurtas metodas eksperimentiškai įvertintas MBSE aplinkoje, taikant jį trijų skirtingų sistemų alternatyvų palyginimo tyrimui atlikti:

- elektrinių traukinio bėgių architektūrų palyginimas pristatytas straipsnyje, išspausdintame mokslo žurnale *IEEE Access* [131];
- jūrų paieškos ir gelbėjimo misijos architektūrų palyginimas pristatytas straipsnyje, išspausdintame konferencijos publikacijoje *INCOSE IS* [134];
- reikalavimų valdymo (RV) sprendimo architektūrų palyginimas išsamiai pristatytas disertacijos tekste. RV sprendimo architektūrų palyginimas detalizuotas toliau.

Alternatyvių RV sprendimo architektūrų palyginimas taikant UT3SA metodą buvo atliktas vertinant aštuonias alternatyvias RV sprendimo architektūras, siekiant atrinkti geriausiai reikalavimus atitinkančią RV architektūrą. Alternatyvų palyginimas apėmė penkis pagrindinius etapus: 1) duomenų paruošimas alternatyvų palyginimui, įskaitant atrankos kriterijų ir alternatyvių RV sprendimo architektūrų specifikavimas MBSE aplinkoje; 2) bazinė patikra; 3) alternatyvių architektūrų įvertinimas pagal atrankos kriterijus; 4) jautrumo analizė; 5) rezultatų peržiūra ir galutinis sprendimas.

Duomenų paruošimo etape buvo aprašoma siekiamo RV sprendimo loginė struktūra, pateikiant reikalingas integracijas su kitais įrankiais. Taip pat nurodyti nauji gebėjimai (angl. *capabilities*) ir proceso veiklos, kuriuos siekiamo RV sprendimo architektūra turi tenkinti. Kritiniai gebėjimai ir proceso veiklos identifikuotos taikant pasiūlytą *Critical* stereotipą. Atrankos kriterijai specifikuoti naudojant *Requirement* elementą su naujai įvestu *PrioritizedRequirement* stereotipu. Kiekvienam atrankos kriterijui nurodytas prioriteto lygis naudojant *PrioritizedRequirement* stereotipo *priority* žymą (angl. *tag*). Iš viso parengti 207 atrankos kriterijai. Kiekvienai alternatyviai RV sprendimo architektūrai buvo sukurtas *FieldedCapability* elementas ir priskirtas atitikimo įvertis pagal kiekvieną atrankos kriterijų.

Bazinės patikros metu buvo atliekami dvi patikros: 1) kritinių reikalavimų atitikimas; 2) SoS architektūrų kokybės įvertinimas. Vykdant kritinių reikalavimų atitikimo tikrinimą, pirmiausia buvo formalizuojami kiekybiniai reikalavimai, siekiant juos verifikuoti automatizuotu būdu pasitelkiant simuliaciją. Kritinių gebėjimų ir procesų veiklų patikrai buvo naudojamasi UT3SA metodu parengtomis *KeyReqCoverage* metrikomis. Alternatyvių architektūrų kokybės įvertinimui buvo naudojamasi UT3SA metodu parengtomis *Quality Assessment* metrikomis. Prieš vykdant kokybės įvertinimą pasirinkta minimali galima kokybės indekso riba – 0,7.

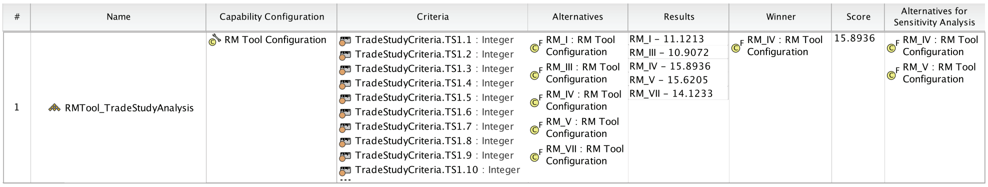


#	Date	Name	Scope	Partial Quality Index_ H	Partial Quality Index_ S Tvs RS	Partial Quality Index_ O Pvs RS	Quality Index
1	2021.12.05 16.56	RM_I_QualityAssessment	Strategy Operational RM_I Resources Structure	0.0769	0.1622	0.15	0.8703
2	2021.12.05 16.58	RM_II_QualityAssessment	Strategy Operational RM_II Resources Structure	0.1519	0.625	0.2612	0.654
3	2021.12.05 16.58	RM_III_QualityAssessment	Strategy Operational RM_III Resources Structure	0.1591	0.0313	0.0795	0.91
4	2021.12.05 16.58	RM_IV_QualityAssessment	Strategy Operational RM_IV Resources Structure	0.0294	0.1176	0.0354	0.9392
5	2021.12.05 16.58	RM_V_QualityAssessment	Strategy Operational RM_V Resources Structure	0.3043	0.038	0.0819	0.8586
6	2021.12.05 16.58	RM_VI_QualityAssessment	Strategy Operational RM_VI Resources Structure	0.0517	0.1786	0.6901	0.6932
7	2021.12.05 16.59	RM_VII_QualityAssessment	Strategy Operational RM_VII Resources Structure	0.0857	0.2083	0.0974	0.8695

6 pav. Kokybės įvertinimo rezultatai MSoSA įrankyje

Atlikus šias dvi patikras buvo pašalintos trys alternatyvios RV sprendimo architektūros. Viena architektūra pašalinta dėl kritinių reikalavimų neatitikimo, kitos dvi pašalintos dėl pasiekto per žemo kokybės indekso (6 pav. Kokybės įvertinimo rezultatai MSoSA įrankyje pav.).

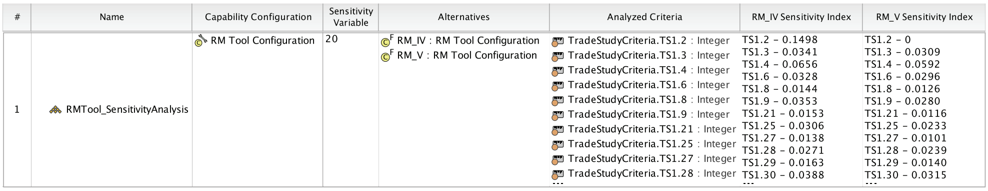
Po bazinės patikros buvo vykdomas alternatyvių architektūrų įvertinimas pagal atrankos kriterijus. Tam buvo naudojama UT3SA metodu pasiūlyta *EvaluationByCriteriaTable* lentelė. Nurodžius šioje lentelėje alternatyvias RV sprendimo architektūras ir atrankos kriterijus, buvo vykdoma sistemų architektūrų simuliacija. Jos rezultatai išsaugoti modelyje ir pavaizduoti lentelėje. RV sprendimo architektūra, kuri gavo didžiausią įvertinimo balą, buvo laikoma „laimėtoja“ (7 pav. Įvertinimo pagal atrankos kriterijus rezultatai MSoSA CASE įrankyje pav.). Taip pat nuspręsta papildomai atlikti jautrumo analizę, nes kita alternatyva gavo labai panašų balą kaip ir „nugalėtoji“ alternatyva (<5 %).



#	Name	Capability Configuration	Criteria	Alternatives	Results	Winner	Score	Alternatives for Sensitivity Analysis
1	RMTool_TradeStudyAnalysis	RM Tool Configuration	TradeStudyCriteria.TS1.1 : Integer TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.5 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.7 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.10 : Integer ...	RM_I : RM Tool Configuration RM_III : RM Tool Configuration RM_IV : RM Tool Configuration RM_V : RM Tool Configuration RM_VII : RM Tool Configuration	RM_I – 11.1213 RM_III – 10.9072 RM_IV – 15.8936 RM_V – 15.6205 RM_VII – 14.1233	RM_IV : RM Tool Configuration	15.8936	RM_IV : RM Tool Configuration RM_V : RM Tool Configuration

7 pav. Įvertinimo pagal atrankos kriterijus rezultatai MSoSA CASE įrankyje

Jautrumo analizė buvo vykdoma pasitelkiant UT3SA metodu pristatytą *SensitivityAnalysisTable* lentelę. Nurodžius jautrumo kintamąjį – 20, buvo vykdoma sistemų architektūrų simuliacija. Jai pasibaigus, lentelėje pateikiami apskaičiuoti jautrumo indeksai kiekvienam atrankos kriterijui (8 pav.).



#	Name	Capability Configuration	Sensitivity Variable	Alternatives	Analyzed Criteria	RM_IV Sensitivity Index	RM_V Sensitivity Index
1	RMTool_SensitivityAnalysis	RM Tool Configuration	20	RM_IV : RM Tool Configuration RM_V : RM Tool Configuration	TradeStudyCriteria.TS1.2 : Integer TradeStudyCriteria.TS1.3 : Integer TradeStudyCriteria.TS1.4 : Integer TradeStudyCriteria.TS1.6 : Integer TradeStudyCriteria.TS1.8 : Integer TradeStudyCriteria.TS1.9 : Integer TradeStudyCriteria.TS1.21 : Integer TradeStudyCriteria.TS1.25 : Integer TradeStudyCriteria.TS1.27 : Integer TradeStudyCriteria.TS1.28 : Integer ...	TS1.2 – 0.1498 TS1.3 – 0.0341 TS1.4 – 0.0656 TS1.6 – 0.0328 TS1.8 – 0.0144 TS1.9 – 0.0353 TS1.21 – 0.0153 TS1.25 – 0.0306 TS1.27 – 0.0138 TS1.28 – 0.0271 TS1.29 – 0.0163 TS1.30 – 0.0388 ...	TS1.2 – 0 TS1.3 – 0.0309 TS1.4 – 0.0592 TS1.6 – 0.0296 TS1.8 – 0.0126 TS1.9 – 0.0280 TS1.21 – 0.0116 TS1.25 – 0.0233 TS1.27 – 0.0101 TS1.28 – 0.0239 TS1.29 – 0.0140 TS1.30 – 0.0315 ...

8 pav. Jautrumo analizės rezultatai MSoSA CASE įrankyje

Ir pagaliau buvo apskaičiuojamas galutinis alternatyvų balas. Galutinis balas parodė, jog alternatyva, kuri buvo pelniusi truputį mažiau taškų po įvertinimo pagal atrankos kriterijus yra pasirenkama kaip „nugalėtoji“ RV sprendimo architektūra. Tokį sprendimą lėmė perskaičiuotas galutinis balas po atliktos jautrumo analizės.

Atlikto alternatyvių RV sprendimo architektūrų palyginimo rezultatai parodė, kad UT3SA metodu pateikiamas procesas ir visi reikiami elementai bei įvertinimo algoritmai, kurie būtini, kai norima atlikti alternatyvų palyginimą MBSE aplinkoje pasitelkiant simuliaciją.

4.1 Eksperimento rezultatų patikimumo ir teisingumo patikra

Patikimumo patikra

Ji atlikta penkis kartus pakartojant alternatyvių RV sprendimo architektūrų palyginimą, kuris pateikiamas 3.2.2 skyriuje, kiekvieną kartą pašalinant vieną ar daugiau alternatyvų iš pradinio alternatyvų rinkinio (Table **3.9** lentelė). Svarbiausias patikimumo patikros tikslas – įsitikinti, jog alternatyvių SoS architektūrų įvertinimo rezultatai yra vienodi, nepaisant vertinamų alternatyvų rinkinio.

Atlikus eksperimento rezultato patikimumo patikrą išaiškėjo, jog buvo gauti tie patys įvertinimo rezultatai, nepaisant pradinio alternatyvų rinkinio, kuris buvo įtrauktas į alternatyvų palyginimo tyrimą. Taigi atlikta patikra patvirtino, jog eksperimento rezultatai yra patikimi.

Teisingumo patikra

Eksperimento rezultatų teisingumo patikra atlikta taikant UT3SA metodą trims eksperimentams: elektrinių traukinio bėgių architektūrų palyginimui (ERC), jūrų paieškos ir gelbėjimo misijos architektūrų palyginimui (MSR) ir RV palyginimui (Table **3.10** lentelė). Eksperimentų rezultatai, kurie buvo gauti taikant

UT3SA metodą, įvertinti ekspertų grupės ir palyginti su anksčiau publikuotais tyrimų rezultatais [142], [52], [143], [144]. Esminis tikslas atliekant eksperimento rezultatų teisingumo patikrą – įsitikinti, jog UT3SA metodas iš galimų alternatyvų atrenka labiausiai pageidaujamą SoS architektūros alternatyvą.

Atlikus eksperimento rezultatų teisingumo patikrą išaiškėjo, jog rezultatai, gauti taikant UT3SA metodą, yra tokie patys, kokių tikėtasi pagal ekspertų grupės vykdytą tyrimą. Atlikta teisingumo patikra patvirtino, jog eksperimento rezultatai yra teisingi.

4.2 UT3SA kokybės pagerinimo įvertinimas atliekant alternatyvų palyginimą

Sukurtu UT3SA metodu siekiama pagerinti alternatyvių SoS architektūrų palyginimo kokybę ir pateikiamos priemonės, leidžiančios automatizuoti SoS architektūrų įvertinimą pasitelkiant sistemos simuliaciją MBSE aplinkoje. Alternatyvų palyginimo kokybės gerinimas reiškia galimų klaidų skaičiaus sumažinimą vertinant alternatyvias SoS architektūras. Taip kartu sumažinama netinkamos alternatyvios SoS architektūros pasirinkimo rizika.

UT3SA metodas apima MBSE, UAF ir automatizavimo / simuliacijos priemones. Literatūroje paskelbta nemažai tyrimų, kurie įrodo ir patvirtina, kad šios priemonės pagerina kokybę sumažindamos galimų klaidų kiekį: MBSE [145], [65]; UAF [146], [147]; automatizavimas / simuliacija [148], [149], [150], [151]. Taip pat atlikta apklausa [152] atskleidė, jog 92 % respondentų teigia, kad automatizavimas atitiko arba viršijo jų lūkesčius; pagerėjo kokybė / tikslumas (90 %), pagerėjo našumas (86 %) ir sumažėjo sąnaudos (59 %).

Papildomai buvo atliktas tyrimas siekiant palyginti klaidų kiekį, kai SoS architektūros yra įvertinamos automatiniu ir rankiniu būdu. Tyrimas buvo suskirstytas į du vertinimus: bazinę patikrą ir detaliąją patikrą. Bazinės patikros metu aptiktų klaidų skaičius SoS architektūrose buvo nustatytas taikant UT3SA metodo *Base Check* žingsnio vertinimo principus. Detaliosios patikros metu, taikant UT3SA metodo *Deep Check* žingsnio vertinimo principus, nustatytas klaidų skaičius, padarytas vertinant atrankos kriterijų atitikimą. Automatinis vertinimas atliktas naudojant UT3SA metodu teikiamas automatizuotas priemones, paremtas simuliacija, o štai rankinį vertinimą atliko ekspertų grupė (Table 3.11 lentelė).

Atliktas tyrimas atskleidė, jog naudojant automatizuotas priemones įvertinti SoS architektūroms bazinės patikros žingsnyje aptinkama mažiausiai 22 % daugiau klaidų nei tą darant rankiniu būdu. Taip pat pastebėta, jog vertinant atrankos kriterijų atitikimą rankiniu būdu, klaidos neišvengiamos. O štai automatizuotas vertinimas pašalina žmogiškojo faktoriaus paliekamas klaidas.

Taip pat svarbu paminėti, jog rastų klaidų skaičius yra glaudžiai susijęs su vertinamų elementų skaičiumi SoS architektūros modeliuose (Fig. 3.26 pav.). Kuo didesnis elementų skaičius, tuo didesnė tikimybė tuose modeliuose rasti klaidų.

5 IŠVADOS

1. ADF analizė atskleidė, jog esami ADF neapibrėžia, kaip turi būti atliekamas SoS architektūrų alternatyvų palyginimas, ir nepateikia formalizuoto alternatyvų palyginimo proceso. ADF leidžia tik specifikuoti reikiamus

duomenis alternatyvų palyginimui ir sudaro sąlygas atlikti SoS architektūros modelių simuliaciją pasitelkiant simuliacijos standartus.

2. MBSE, UAF ir fUML analizė atskleidė, kad tai yra tinkamos priemonės siekiant automatizuoti alternatyvių SoS architektūrų įvertinimus alternatyvų palyginimo procese. Tačiau UAFP metamodelio plėtiniai yra būtini norint atlikti alternatyvų palyginimą MBSE aplinkoje, kadangi jame trūksta konkrečių konceptų, leidžiančių apibrėžti bei atlikti alternatyvų palyginimą MBSE aplinkoje kartu su UAF ir fUML.
3. Alternatyvų palyginimo procesų analizė atskleidė, jog dauguma esamų alternatyvų palyginimo procesų: 1) nesuteikia formalumo, reikalingo alternatyvų palyginimo procesui; 2) apibrėžia tik pagrindinę alternatyvų palyginimo eigą, neįtraukiant alternatyvų įvertinimo detalių, tokių kaip konkretūs įvertinimo metodai, technikos, algoritmai; 3) nėra pritaikyti konkrečiai sričiai, todėl juose pateikiami tik bendrieji veiksmai ir vartojama bendroji terminologija; 4) nepateikia jokių gairių, kaip taikyti alternatyvų palyginimo procesą MBSE aplinkoje; 5) neapibrėžia procese dalyvaujančių vaidmenų ir nenurodo kiekvieno etapo įvesties / išvesties duomenų. Kadangi UAF yra OMG standartas, apimantis visas reikiamas charakteristikas apibrėžti sudėtingų sistemų architektūroms, jis gali būti naudojamas kaip pagrindas alternatyvų palyginimo procesui formalizuoti. Tai galėtų padėti sumažinti paminėtus esamų alternatyvų palyginimo procesų trūkumus. Be to, UAF suteikia galimybę taikyti simuliacija grįstus metodus architektūrų įvertinimui.
4. Remiantis atlikta analize, buvo suformuluotas naujas alternatyvų palyginimo metodas. Šitas metodas (UT3SA) dabar yra vienintelis metodas, apimantis visas šias savybes: 1) pateikia formaliai apibrėžtą alternatyvų palyginimo procesą su apibrėžtais vaidmenimis, įvesties ir išvesties duomenis; 2) pateikia gaires, kaip apibrėžti alternatyvų palyginimą MBSE aplinkoje; 3) nurodo išsamius sistemos architektūros įvertinimo metodus, įskaitant vertinimo algoritmus ir taisykles; 4) suteikia priemones atlikti ankstyvąjį architektūros kokybės įvertinimą.
5. UT3SA metodas įgyvendintas kaip UAF profilis su alternatyvų palyginimo stereotipais (UT3SA profilis) ir kaip paketas, pateikiantis naujas lenteles, validacijos taisykles, metrikas bei algoritmus, skirtus vykdyti automatizuotam alternatyvių architektūrų įvertinimui MBSE aplinkoje. UT3SA metodo taikymas leidžia tiesiogiai apibrėžti reikiamus alternatyvų palyginimo duomenis modelyje, nustatyti alternatyvių sistemų architektūrų kokybę ankstyvame alternatyvų palyginimo etape, įvertinti alternatyvias SoS sprendimo architektūras pagal atrankos kriterijus ir atlikti jautrumo analizę, siekiant išsirinkti tinkamiausią sprendimų architektūrą. UT3SA profilis ir UT3SA paketas yra sukurti naudojant *Catia MSoSA v2021xRefresh2* įrankį ir gali būti įdiegti bet kuriame su UAF suderinamame CASE įrankyje.
6. UT3SA metodas eksperimentiškai taikytas atliekant aštuonių alternatyvių RV sprendimo architektūrų palyginimą. Alternatyvos buvo lyginamos

keturiais etapais: 1) kritinių reikalavimų atitikties patikra; 2) kokybės įvertinimas; 3) alternatyvių sprendimų architektūrų įvertinimas pagal atrankos kriterijus; 4) jautrumo analizė. Eksperimentinis vertinimas įrodė, jog UT3SA metodas suteikia: 1) būtiną UAFP išplėtimą, kuris leidžia tiesiogiai apibrėžti alternatyvų palyginimą MBSE aplinkoje; 2) detalų alternatyvų palyginimo procesą, kuris apibrėžia konkrečius etapus, įvesties / išvesties duomenis bei įvertinimo metodus, ir tai leidžia atlikti išsamų alternatyvių SoS architektūrų palyginimą; 3) būtinas sistemos architektūros įvertinimo taisykles ir algoritmus, siekiant pasirinkti geriausiai reikalavimus atitinkančią sistemos architektūrą iš kelių galimų alternatyvų. Taip pat atliktas eksperimentinis vertinimas patvirtina, jog sukurtas UT3SA metodas gali būti taikomas praktiškai.

7. Atliktas eksperimento rezultatų patikimumo ir teisingumo patikra atskleidė bei patvirtino, jog eksperimento rezultatai yra patikimi ir teisingi. Tad galima teigti, jog UT3SA metodu atrenkama labiausiai pageidaujama SoS architektūros alternatyva iš galimų.
8. UT3SA metodo kokybės pagerinimo vertinimas atskleidė, jog automatizuotas SoS architektūrų vertinimas nustato mažiausiai 22 % daugiau klaidų nei vertinimas rankiniu būdu. Klaidų nustatymo atvejų kiekis susijęs su SoS architektūroje esančių elementų skaičiumi. Kuo daugiau elementų, tuo didesnis automatizuotomis priemonėmis aptinkamų klaidų procentinis skirtumas lyginant su rankiniu vertinimu. Taip pat atlikta MBSE, UAF ir simuliacijos priemonių analizė patvirtino jų daromą įtaką kokybės gerinimui mažinant klaidų skaičių. Papildomai Deloitte atlikta apklausa atskleidė, jog net 92 % respondentų patvirtino, kad automatizavimas ir simuliacija atitiko arba viršijo jų lūkesčius, todėl pagerėjo kokybė / tikslumas (90 %).

REFERENCES

[1] P. J. Baker and J. T. Whalen, "Survey of Trade Study Methods for Practical Decision-Making," in *Annu. NASA IV&V Workshop*, 2012.

[2] K. Henderson and A. Salado, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Systems Engineering,* vol. 24, no. 1, pp. 51-66, 2021.

[3] INCOSE, "Systems Engineering Vision 2035," 2021. [Online]. Available: https://www.incose.org/docs/default-source/se-vision/incose-se-vision-2035.pdf?sfvrsn=e32063c7_4.

[4] S. Gao, W. Cao, L. Fan and J. Liu, "MBSE for Satellite Communication System Architecting," *IEEE Access,* vol. 7, pp. 164051-164067, 2019.

[5] L. Li, N. L. Soskin, A. Jbara, M. Karpel and D. Dori, "Model-Based Systems Engineering for Aircraft Design With Dynamic Landing Constraints Using Object-Process Methodology," *IEEE Access,* vol. 7, pp. 61494-61511, 2019.

[6] E. Brusa, "Digital Twin: Toward the Integration Between System Design and RAMS Assessment Through the Model-Based Systems Engineering," *IEEE Systems Journal,* vol. 15, no. 3, pp. 3549-3560, 2021.

[7] International Organization for Standardization, *ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary, edition 2,* Geneva, 2017.

[8] D. Harris, J. Bonometti, R. Mueller and G. Murphy, "Lunar Engineering Handbook," [Online]. Available: http://www.eng.auburn.edu/~dbeale/ESMDCourse/Chapter4.htm.

[9] Office of Aerospace Studies, "Analysis of Alternatives (AoA) Handbook, A Practical Guide to the Analysis of Alternatives," 2016. [Online]. Available: https://afacpo.com/AQDocs/AoAHandbook.pdf.

[10] M. M. LaSorda, J. Borky and R. Sega, "Model-based architecture optimization for major acquisition analysis of alternatives," in *IEEE Aerospace Conference*, Big Sky, MT, USA, 2018.

[11] M. M. LaSorda, J. Borky and R. Sega, "Model-Based Systems Architecting with Decision Quantification for Cybersecurity, Cost, and Performance," in *IEEE Aerospace Conference*, 2020.

[12] W. Stecz and P. Kowaleczko, "Designing Operational Safety Procedures for UAV According to NATO Architecture Framework," in *16th International Conference on Software Technologies* , 2021.

[13] J.E. Hannay, "Architectural work for modeling and simulation combining the NATO Architecture Framework and C3 Taxonomy," *Journal of Defense Modeling and Simulation,* vol. 14, no. 2, pp. 139-158, 2017.

[14] C. Zhiwei, Z. Tingdi, J. Jian and L. Yaqiu, "System of Systems Architecture Modeling and Mission Reliability Analysis Based on DoDAF and Petri Net,"

in *Annual Reliability and Maintainability Symposium (RAMS)*, Orlando, USA, 2019.

[15] Y. Tong, J. Zhang, M. Xu and T. Qin, "Network Security Monitoring and Defense System Framework Design Using Mobile Agents Based on DoDAF," in *International Conference on Computer Science and Applications (CSA)*, Wuhan, China, 2015.

[16] Q. Xiong, G. Chen, L. Chang, T. Liao and Z. Mao, "System of systems supportability assessment model based on DoDAF," in *29th Chinese Control And Decision Conference (CCDC)*, Chongqing, China, 2017.

[17] OMG, "Unified Architecture Framework (UAF) 1.0," 2017. [Online]. Available: https://www.omg.org/spec/UAF/1.0/.

[18] A. Morkevicius, J. Bankauskaite and N. Jankevicius, "Towards Standards-based Execution of System of Systems Models," in *5th International Conference of System of Systems Engineering (SoSE)*, Budapest, Hungary, 2020.

[19] J. Towers, "The MBSE Horizon: Advances and Challenges over the next few years," in *PPI SyEN 85*, 2020.

[20] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly,* vol. 28, no. 1, pp. 75-105, 2004.

[21] United States Air Force Scientific Advisory Board, "Report on System of Systems Engineering for Air Force Capability Development," SAB-TR-05-04, 2005.

[22] BKCASE, "Body of Knowledge and Curriculum to Advance Systems Engineering," 2018. [Online]. Available: https://www.sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK).

[23] D. Kemp and E. Evans, "Steampunk System of Systems Engineering: A case study of successful System of Systems engineering in 19th century Britain," *Insight,* vol. 19, no. 3, pp. 27-29, 2016.

[24] M. Henshaw, "Systems of Systems. Cyber-Physical Systems, the Internet of Things… Whatever Next," *Insight,* vol. 19, no. 3, pp. 51-54, 2016.

[25] H. Liu, Y. Tian, Y. Gao, J. Bai and J. Zheng, "System of systems oriented flight vehicle conceptual design: Perspectives and progresses," *Chinese Journal of Aeronautics,* vol. 28, no. 3, pp. 617-635, 2015.

[26] G. Fortino, C. Savaglio, G. Spezzano and M. Zhou, "Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools," *IEEE Transactions on Systems,* vol. 51, no. 1, pp. 223-236, 2021.

[27] International Organization for Standardization, "ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes, Annex G," Geneva, 2015.

[28] U.S. Department of Defense, "Guide to systems engineering for systems of systems," 2008. [Online]. Available: https://acqnotes.com/wp-

content/uploads/2014/09/DoD-Systems-Engineering-Guide-for-Systems-of-Systems-Aug-2008.pdf.

[29] Defense Acquisition University, "Defense acquisition guidebook," 28 July 2011. [Online]. Available: https://www.dau.edu/pdfviewer?Guidebooks/DAG/DAG-CH-3-Systems-Engineering.pdf.

[30] M. Jamshidi, "Introduction to system of systems," in *System of Systems Engineering: Principles and Applications*, Boca Raton, CRC Press, 2009, pp. 1-37.

[31] M. Maier, "Architecting principles for systems-of-systems," *Systems Engineering,* vol. 1, no. 4, p. 267–284, 1998.

[32] INCOSE, "About Incose," [Online]. Available: https://www.incose.org/about-incose.

[33] U. S. Department of Defense, [Online]. Available: https://www.defense.gov/About/.

[34] Federal geographic data committee, "Who we are," [Online]. Available: https://www.fgdc.gov/who-we-are/history. [Accessed].

[35] NIST, "Systems Security Engineering Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems," *SP 800-160,* vol. 1, 2016.

[36] MITRE, "System Engineering Guide," [Online]. Available: https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/architectural-frameworks-models-and-views.

[37] The Open Group Standard, The TOGAF Standard, Version 9.2, No C182, Van Haren Publishing; 11th edition, 2018.

[38] Department of Defense Office of the Assistant Secretary of Defense (OASD) for Network Infrastructure and Integration, "The DoDAF Architecture Framework Version 2.02," 2010.

[39] Department of Defense, "DoD Architecture Framework Version 2.02, Manager's Guide," 2015.

[40] North Atlanic Treaty Organization, "NATO Architecture Framework, Version 4," 2018. [Online]. Available: https://www.nato.int/cps/en/natohq/topics_157575.htm.

[41] Minestery of Defense, "Guidance MOD Architecture Framework," 2012. [Online]. Available: https://www.gov.uk/guidance/mod-architecture-framework.

[42] OMG, "Unified Architecture Framework Profile (UAFP) version 1.0," 2017. [Online]. Available: https://www.omg.org/spec/UAF/1.0/PDF.

[43] OMG, "Unified Architecture Framework (UAF), The Domain Metamodel, Version 1.0 - Appendix A," 2017. [Online]. Available: https://www.omg.org/cgi-bin/doc?formal/17-12-02.pdf.

[44] Executive office of the president of the united states, "FEA Consolidated Reference Model Document Version 2.3," 2007. [Online]. Available: https://www.reginfo.gov/public/jsp/Utilities/FEA_CRM_v23_Final_Oct_2007_Revised.pdf.

[45] The Open Group, "The Open Group Architectural Framework," 2015. [Online]. Available: www.opengroup. org /architecture/togaf7-doc/arch/.

[46] OMG, "Unified Architecture Framework (UAF) 1.1," 2020. [Online]. Available: https://www.omg.org/spec/UAF/1.1/About-UAF/.

[47] A. Morkevicius, "Business and information systems alignment method based on enterprise architecture models," Kaunas, 2013.

[48] O. C. Eichmann, S. Melzer and R. God, "Model-based Development of a System of Systems Using Unified Architecture Framework (UAF): A Case Study," in *IEEE International Systems Conference (SysCon)*, 2019.

[49] OMG, "Unified Profile for DoDAF and MODEM (UPDM V3.0) Request For Proposal, c4i/2013-09-11," 2013.

[50] P. Barnes, "Utilizing MBSE to Modularly Architect the NESDIS Ground Enterprise," in *14th Annual Symposium on New Generation Operational Environmental Satellite Systems*, 2018.

[51] F. Dandashi and M. C. Hause, "UAF for system of systems modeling," in *10th System of Systems Engineering Conference (SoSE)*, San Antonio, 2015.

[52] L. O. Kihlström, I. Karlsson and B. Chen, "How to use Architecture frameworks to speed up systems development," in *Nordic Systems Engineering*, 2018.

[53] P. Sjöberg, L. O. Kihlström and M. Hause, "An industrial example of using Enterprise Architecture to speed up systems development," in *27th INCOSE International Symposium*, 2017.

[54] OMG, "Unified Architecture Framework Profile (UAFP), v1.1," 2020. [Online]. Available: https://www.omg.org/spec/UAF/1.1/UAFP/PDF.

[55] L. Tirone, C. Agostinelli, P. Petrinca, E. Guidolotti, L. Fornaro, M. Nardini and S. M. Solazzi , "Application of the Unified Architecture Framework for the Definition of a Generic System Architecture of a Combat System," in *3rd INCOSE Italia Conference on Systems Engineering*, Naples, Italy, 2017.

[56] J. Bankauskaite, A. Morkevicius and R. Butleris, "Early Quality Evaluation of System of Systems Architecture Within Trade Study Process," *IEEE Access,* vol. 8, pp. 220858-220868, 2020.

[57] J. Bankauskaite and A. Morkevicius, "Towards an Automated UAF-based Trade Study Process for System of Systems Architecture," in *INCOSE International Symposium*, Cape Town, South Africa, 2020.

[58] R. Karban, M. Zamparelli, B. Bauvir and G. Chiozzi, "Three years of MBSE for a large scientific programme: Report from the Trenches of Telescope Modeling," in *INCOSE International Symposium*, Rome, Italy, 2012.

[59] J. D'Ambrosio and G. Soremekun, "Systems engineering challenges and MBSE opportunities for automotive system design," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2017.

[60] OMG, "MBSE specifications at OMG," [Online]. Available: https://www.omg.org/intro/MBSE.pdf.

[61] NASA, "Pushing the State of the Art: A Web-enabled MBSE Analysis Integration Framework, Phase I," Space Technology Mission Directorate, 2020.

[62] E. Carroll and R. Malins, "Systematic Literature Review: How is Model-Based Systems Engineering Justified," Sandia Report SAND106-2607, Sandia National Laboratory, New Mexico, 2016.

[63] K. Henderson and A. Salado, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Systems engineering,* vol. 24, no. 1, pp. 51-66, 2020.

[64] J. B. Holladay, J. Knizhnik, K. J. Weiland, A. Stein, T. Sanders and P. Schwindt, "MBSE Infusion and Modernization Initiative (MIAMI): “Hot” Benefits for Real NASA Applications," in *IEEE Aerospace Conference*, 2019.

[65] T. Bayer, "Is MBSE helping? Measuring value on Europa Clipper," in *IEEE Aerospace Conference*, 2018.

[66] M. Chami and J.M. Bruel, "A Survey on MBSE Adoption Challenges," in *The Systems Engineering Conference of the Europe, Middle-East and Africa (EMEA) Sector of INCOSE*, Berlin, Germany, 2018.

[67] S. Huda, S. Alyahya, M. Mohsin Ali, S. Ahmad, J. Abawajy, H. Al-Dossari and J. Yearwood, "A Framework for Software Defect Prediction and Metric Selection," *IEEE Access,* vol. 6, pp. 2844-2858, 2018.

[68] Y. U. Mshelia, S. T. Apeh and E. Olaye, "Towards a Unified Process Model for Comprehensive Software Metrics Suite: An Introduction," in *19th International Conference on Computational Science and Its Applications (ICCSA)*, Saint Petersburg, Russia, 2019.

[69] D. Lübke, "Using Metric Time-Lines for Identifying Architecture Shortcomings in Process Execution Architectures," in *IEEE/ACM 2nd International Workshop on Software Architecture and Metrics*, Florence, Italy, 2015.

[70] J. Schumann and K. Goseva-Popstojanova, "Verification and Validation Approaches for Model-Based Software Engineering," in *ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Munich, Germany, 2019.

[71] A. Morkevicius and N. Jankevicius, "An approach: SysML-based automated requirements verification," in *IEEE International Symposium on Systems Engineering (ISSE)*, Rome, Italy, 2015.

[72] Sparx Systems, 2022. [Online]. Available: https://sparxsystems.com/products/mdg/index.html#MDGI.

[73] Unicom, "System Architect," 2022. [Online]. Available: https://www.teamblue.unicomsi.com/products/system-architect/.

[74] PTC, "SysML, UAF, UML, and UPDM Profiles," 2022. [Online]. Available: https://support.ptc.com/help/modeler/r9.0/en/index.html#page/Integrity_Modeler%2Fsysml%2FWhats_new_in_the_62_patch_bdd.html%23.

[75] Dassault Systemes, "Magic Systems of Systems Architect 2021xR2," 2022. [Online]. Available: https://docs.nomagic.com/display/MSOSA2021xR2/Magic+Systems+of+Systems+Architect+Documentation.

[76] K. Azad, S. Sint, G. Gessl, K. Zeman, F. Jungreitmayr, H. Wahl, A. Wenigwieser and R. Kretschmer, "Towards a logical framework for ideal MBSE tool selection based on discipline specific requirements," *Journal of Systems and Software,* vol. 189, no. 2, 2022.

[77] OMG, "Semantics of a Foundational Subset for Executable UML Models (fUML)," 2021.

[78] OMG, "Precise Semantics of UML Composite Structure (PSCS), Version 1.2," 2019. [Online]. Available: https://www.omg.org/spec/PSCS/1.2.

[79] OMG, "Precise Semantics of UML State Machines (PSSM), Version 1.0," 2019. [Online]. Available: https://www.omg.org/spec/PSSM/About-PSSM/.

[80] H. Husain Arifin, Y. Dong, R. Ong, Y. Gu, N. Chimplee and D. Wu, "Hatley-Pirbhai Control Flow Diagram with SysML for Early Validation," in *INCOSE International Symposium*, 2020.

[81] P. Weyprecht and O. Rose, "Model-driven Development of Simulation Solution based on SysMLstarting with the Simulation Core," in *Spring Simulation Multi-conference, SpringSim*, Boston, MA, USA, 2011.

[82] D. M. Buede and W. D. Miller, The engineering design of systems: Models and methods, 3rd edition, Wiley, 2016.

[83] National Airspace System, "NAS System engineering manual, Version 3.1," 2006.

[84] Defense systems management college, "Systems engineering fundamentals," 2016.

[85] NASA, "Expanded Guidance for NASA Systems Engineering, Volume 1: Systems engineering practices," 2016. [Online]. Available: https://ntrs.nasa.gov/api/citations/20170007238/downloads/20170007238.pdf .

[86] J. Reiter, "Numerical and analytical solutions to rapid collision avoidance maneuvers constrained by mission performance requirements," in *MS in Aerospace Engineering*, 2016.

[87] R. Sabetto and L. Gately, "The trade study process," 2021. [Online]. Available: https://www.mitre.org/sites/default/files/publications/pr-21-0522-the-trade-study-process.pdf.

[88] NASA, "NASA Systems Engineering Handbook," Washington: 12th Media Services, 2017.

[89] Z. Zhang and S. Balakrishnan, "Multi-Criteria Decision Analysis," *International Encyclopedia of Transportation,* pp. 485-492, 2021.

[90] E. S. Mityakov, D. N. Lapaev, M. I. Beskhmelnov, A. I. Ladynin and N. M. Shmeleva, "Knowledge-intensive Enterprises Multi-criteria Decision-making Energy Security Management Information System," in *Conference of Russian Young Researchers in Electrical and Electronic Engineering* , 2022.

[91] T. Chen, "A Mixed-Choice-Strategy-Based Consensus Ranking Method for Multiple Criteria Decision Analysis Involving Pythagorean Fuzzy Information," *IEEE Access,* vol. 6, pp. 79174-79199, 2018.

[92] S. Jansen, "The Multi-attribute Utility Method," 2011.

[93] D. Von Winterfeldt and W. Edwards, Decision analysis and behavioral research, Cambridge: Cambridge University Press, 1986, p. 273.

[94] R. L. Keeney and H. Raiffa, Decisions with multiple objectives: Preferences and value tradeoffs, New York: Wiley, 1976.

[95] S. Pugh, "Concept selection: a method that works," in *International conference on engineering design*, Rome, 1981.

[96] K. Wurthmann, "Conducting Pugh Method-based Trade Studies during Product Development: The case of evaluating Turbofan versus Turboprop versus Piston Engine Alternatives for UAVs," in *AIAA Scitech 2020 Forum*, Orlando, FL, 2020.

[97] M. S. F. Hussin, M. A. Shamsuddin, R. Jumaidin, A. A. Zakaria and N. Jenal, "Portable Grease Trap for Wastewater Management System: A Conceptual Design Approach," *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences,* vol. 49, no. 1, pp. 18-24, 2020.

[98] Y. Wijnia, D. Clausing, P. Herder, K. Katsikopoulos, E. Subrahmanian and D. Frey, "The Pugh Controlled Convergence Method: Model-Based Evaluation and Implications for Design Theory," *Research in Engineering Design,* vol. 20, pp. 41-58, 2009.

[99] I. Olkin and A.R. Sampson, "Multivariate Analysis: Overview," *International Encyclopedia of the Social & Behavioral Sciences,* 2001.

[100] S. Grimnes and O. G. Martinsen, "Data and Models," *Bioimpedance and Bioelectricity Basics (Third Edition),* 2015.

[101] F. Bensalma, G. Richardson, Y. Ouakrim, A. Fuentes, M. Dunbar, N. Hagemeister and N. Mezghani, "Graphical-based multivariate analysis for knee joint clinical and kinematic data correlation assessment," in *42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada, 2020.

[102] N. Al-Zubi, L. Momani, A. Al-kharabsheh and W. Al-Nuaimy, "Multivariate analysis of intracranial pressure (ICP) signal using principal component analysis," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, MN, USA, 2009.

[103] P. Ren, J. F. Bosch Bayard, L. Dong, J. Chen, L. Mao, D. Ma, M. A. Sanchez, D. M. Morejon, M. L. Bringas, D. Yao, M. Jahanshahi and P. A. Valdes-Sosa, "Multivariate Analysis of Joint Motion Data by Kinect: Application to Parkinson's Disease," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 28, no. 1, pp. 181-190, 2020.

[104] H. Taherdoost, "Decision Making Using the Analytic Hierarchy Process (AHP); A Step by Step Approach," *International Journal of Economics and Management System, IARAS,* 2017.

[105] S. Halim, Felecia, D. Wulandari and F. L. Susanti, "Group decision using analytical hierarchical process: Surabaya's universities library in digital natives perspective," in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Bali, Indonesia, 2016.

[106] A. A. B. Abdelnabi, "An Analytical Hierarchical Process Model to Select Programming Language for Novice Programmers for Data Analytics Applications," in *nternational Arab Conference on Information Technology (ACIT)*, Al Ain, United Arab Emirates, 2019.

[107] S. Yusoh and S. Matayong, "Heuristic evaluation of online satisfaction survey system for public healthcare service: Applying analytical hierarchical process," in *nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2017.

[108] S. Oguztimur, "Why fuzzy analytic hierarchy process approach for transport problems?," *Research Papers in Economics,* 2011.

[109] R. Karthikeyan, K. Venkatesan and A. Chandrasekar, "A Comparison of Strengths and Weaknesses for Analytical Hierarchy Process," *Journal of Chemical and Pharmaceutical Sciences,* vol. 9, no. 3, 2019.

[110] C. L. Hwang and K. Yoon, Multiple Attribute Decision Making: Methods and Applications, New York: Springer-Verlag, 1981.

[111] S. Chakraborty, "TOPSIS and Modified TOPSIS: A comparative analysis," *Decision analytics journal,* vol. 2, 2022.

[112] M. Amudha, M. Ramachandran, V. Saravanan, P. Anusuya and R. Gayathri, "A Study on TOPSIS MCDM Technies and its Application," *Data Analytics and Artificial Intelligence,* vol. 1, no. 1, pp. 9-14, 2021.

[113] M. El Alaoui, "TOPSIS Methodology and Limits," in *Fuzzy TOPSIS*, Boca Raton, CRC Press, 2021, pp. 30-38.

[114] Z. Yishang, L. Yongshou and Y. Xufeng, "Mathematical Applications to Reliability and Maintenance Problems in Engineering Systems," *Mathematical Problems in Engineering,* pp. 1-13, 2015.

[115] D. Fitzpatrick, "Sensitivity Analysis," in *Analog Design and Simulation Using OrCAD Capture and PSpice (Second Edition)*, Newnes, 2018, pp. 351-365.

[116] C. J. Calderón and J. Ancheyta, "Modeling, simulation, and parametric sensitivity analysis of a commercial slurry-phase reactor for heavy oil hydrocracking," *Fuel,* vol. 244, pp. 258-268, 2019.

[117] A. Anand, M. Agrawal, N. Bhatt and M. Ram, "Software Patch Scheduling Policy Incorporating Functional Safety Standards," in *Advances in System Reliability Engineering*, Academic Press, 2019, pp. 267-279.

[118] D. V. Likhachev, "Parametric sensitivity analysis as an essential ingredient of spectroscopic ellipsometry data modeling: An application of the Morris screening method," *Journal of Applied Physics,* vol. 126, no. 18, 2019.

[119] L. S. Lacerda, P. R. Junior, R. S. Peruchi, G. Chicco, L. C. S. Rocha, G. Aquila and L. M. C. Junior, "Microgeneration of Wind Energy for Micro and Small Businesses: Application of ANN in Sensitivity Analysis for Stochastic Economic Feasibility," *IEEE Access,* vol. 8, pp. 73931-73946, 2020.

[120] W. Jin, Y. Li, X. Lin, X. Guo, X. Cui, H. Liu and J. Fan, "Sensitivity analysis for decision-making of UHV long distance power transmission site selection under uncertainty based on cloud model," in *5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, 2015.

[121] M. T. Kassis, D. Tannir, R. Toukhtarian and R. Khazaka, "Moments-Based Sensitivity Analysis of X-Parameters with respect to Linear and Nonlinear Circuit Components," in *IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2019.

[122] X. Y. Zhang, M. N. Trame, L. J. Lesko and S. Schmidt, "Sobol Sensitivity Analysis: A Tool to Guide the Development and Evaluation of Systems Pharmacology Models," *CPT Pharmacometrics Syst Pharmacol,* vol. 4, no. 2, pp. 69-79, 2015.

[123] E. D. Şandru, E. David and G. Pelz, "Machine Learning-Based Local Sensitivity Analysis of Integrated Circuits to Process Variations," in *EEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020.

[124] J. Zhang, "On The Local Sensitivity Analysis for Phase Expansion," in *Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, 2021.

[125] A. Bucciarelli, A. Adami, C. R. Chandaiahgari and L. Lorenzelli, "Multivariable optimization of inkjet printing process of Ag nanoparticle ink

on Kapton," in *EEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*, 2020.

[126] Reinhard, Stijn & Naranjo, María & Polman, Nico & Hennen, Wil., "Modelling choices and social interactions with a threshold public good: Investment decisions in a polder in Bangladesh," *Land Use Policy,* vol. 113, 2021.

[127] D. M. Hamby, "Comparison of sensitivity analysis techniques," *Health physics,* vol. 68, no. 2, pp. 195-204, 1995.

[128] G. A. Rios, J. S. Rincón Tabares, A. Montoya, D. Restrepo and H. Millwater, "Transient thermomechanical sensitivity analysis using a complex-variable finite element method," *Journal of Thermal Stresses,* vol. 45, no. 5, pp. 341-374, 2022.

[129] F. O. Hoffman and R. H. Gardner, "Radiological Assessments: A Textbook on Environmental Dose Assessment. US Nuclear Regulatory Commission, Washington D.C., Report no. NUREG/CR-3332," 1983, pp. 11.1-11.55.

[130] O. Bo and L. Guangsheng, "Optimization of Measuring Points of Fire Suppression and Suppression Circuit Board Based on Global Sensitivity Analysis," in *EEE 2nd International Conference on Electronics and Communication Engineering (ICECE)*, 2019.

[131] J. Bankauskaite, A. Morkevicius and R. Butleris, "Model-Based Evaluation of the System of Systems Architectures Used to Perform Trade Studies and Sensitivity Analyses," *IEEE Access,* vol. 9, no. 114609-114621, 2021.

[132] A. Morkevicius, L. Bisikirskiene and G. Bleakley, "Using a systems of systems modeling approach for developing Industrial Internet of Things applications," in *12th System of Systems Engineering Conference (SoSE)*, Waikoloa, HI, 2017.

[133] INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition, San Diego, CA (US): Wiley, 2015.

[134] J. Bankauskaite, Z. Strolia and A. Morkevicius, "Automated trade study analysis based on dynamic requirements verification in the model-based system engineering," in *INCOSE international symposium*, Honolulu, HI, USA, 2021.

[135] P. W. Thompson, "The development of the concept of speed and its relationship to concepts of rate," *The Development of Multiplicative Reasoning in the Learning of Mathematics,* pp. 181-234, 1994.

[136] MIPS, "Participating in the Quality Performance Category in the 2022 Performance Year: Traditional MIPS," *EyeNet,* 2022.

[137] Kamal M.Al-Subhi Al-Harbi, "Application of the AHP in project management," *International Journal of Project Management,* vol. 19, no. 1, pp. 19-27, 2001.

[138] Swift A, Heale R, Twycross A., "What are sensitivity and specificity?," *Evidence-Based Nursing,* vol. 23, no. 1, 2020.

[139] Wolfe, C. R., Reyna, V. F., and Smith, R. J., "On Judgments of Approximately Equal," *J. Behav. Dec. Making,* vol. 31, no. 1, p. 151– 163, 2018.

[140] CHUNG-HORNG LUNG and KALAI KALAICHELVAN, "AN APPROACH TO QUANTITATIVE SOFTWARE ARCHITECTURE SENSITIVITY ANALYSIS," *International Journal of Software Engineering and Knowledge Engineering,* vol. 10, no. 1, pp. 97-114, 2000.

[141] J. Knittel, K. Hughes, J. Englander and B. Sarli, "Automated Sensitivity Analysis of Interplanetary Trajectories for Optimal Mission Design," *Trans. JSASS Aerospace Tech. Japan,* vol. 14, no. 31, pp. 1-8, 2017.

[142] Håkan Sundelin, Stefan Tongur, "Architectural description of ERS: Analyzing implications of short and long electric road segments," in *3rd Electric Road Systems Conference*, Frankfurt, Germany, 2019.

[143] Seilevel, "Requirements Management Tool Evaluation Report," Austin, TX, 2016.

[144] Seilevel, "Seilevel's Evaluations of Requirements Management Tools: Summaries and Scores," Austin, TX, 2016.

[145] Henderson, Kaitlin & Salado, Alejandro, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Systems Engineering,* vol. 24, no. 1, pp. 51-66, 2021.

[146] P. Barnes, "Utilizing MBSE to Modularly Architect the NESDIS Ground Enterprise," in *14th Annual Symposium on New Generation Operational Environmental Satellite Systems*, 2018.

[147] Dandashi, F., Hause, M. C., "UAF for system of systems modeling," in *10th System of Systems Engineering Conference (SoSE)*, San Antonio, 2015.

[148] Anand Singh Rajawat, Romil Rawat, Kanishk Barhanpurkar, Rabindra Nath Shaw, Ankush Ghosh, "Robotic process automation with increasing productivity and improving product quality using artificial intelligence and machine learning," in *Artificial Intelligence for Future Generation Robotics*, Elsevier, 2021, pp. 1-13.

[149] Comp, Geoffrey & Silver, Benjamin & Elliott, John & Kalnow, Andrew, "Utilization of Simulation Techniques to Enhance Quality Improvement Processes in the Emergency Department," *Cureus,* vol. 12, no. 2, 2020.

[150] Gogg, T.J. & Mott, J.R.A., "Improve Quality and Productivity with Simulation," 1992.

[151] Mohan Mano Hassan, Fnu & Kalamraju, Sai Priyanka & Dangeti, Venkata Sandeep & Pudipeddi, Sirisha & Williams, Edward, "SIMULATION IMPROVES EFFICIENCY AND QUALITY IN SHOE MANUFACTURING," in *European Modelling and Simulation Symposium*, 2019.

[152] D. LLP, "Robotic process automation (RPA) trends and to-do list for scaling across the enterprise - The 3rd Annual Global RPA Survey Report," The Creative Studio, London, 2022.

LIST OF SCIENTIFIC ARTICLES ON THE DISSERTATION TOPIC

Articles Indexed in the Web of Science with Impact Factor – International Publishers:

1. Bankauskaite, Jovita; Morkevicius, Aurelijus; Butleris, Rimantas. Model-based evaluation of the system of systems architectures used to perform trade studies and sensitivity analyses // IEEE access. Piscataway, NJ: IEEE. ISSN 2169-3536. 2021, vol. 9, p. 114609-114621. DOI: 10.1109/ACCESS.2021.3105589.
2. Bankauskaite, Jovita; Morkevicius, Aurelijus; Butleris, Rimantas. Early quality evaluation of system of systems architecture within trade study process // IEEE Access. Piscataway, NJ: IEEE. ISSN 2169-3536. 2020, vol. 8, p. 220858-220868. DOI: 10.1109/ACCESS.2020.3043036.

Articles Indexed in the Web of Science or Scopus without Impact Factor or SNIP – International Publishers:

1. Morkevicius, Aurelijus; Bankauskaite, Jovita; Jankevicius, Nerijus. Towards standards-based execution of system of systems models // SOSE 2020: IEEE 15th international conference of system of systems engineering, Budapest, Hungary, June 2-4, 2020: proceedings. Piscataway, NJ: IEEE, 2020. ISBN 9781728180519. eISBN 9781728180502. p. 157-162. DOI: 10.1109/SoSE50414.2020.9130510.

Articles in conference proceedings – International Publishers:

1. Bankauskaite, Jovita; Strolia, Zilvinas; Morkevicius, Aurelijus. Automated trade study analysis based on dynamic requirements verification in the model-based system engineering // INCOSE international symposium: 31st annual INCOSE international symposium, Honolulu, HI, USA, July 17-22, 2021. Hoboken, NJ: John Wiley & Sons. ISSN 2334-5837. 2021, vol. 31, iss. 1, p. 1-15. DOI: 10.1002/j.2334-5837.2021.00899.x.
2. Bankauskaite, Jovita; Morkevicius, Aurelijus. Towards an automated UAF-based trade study process for system of systems architecture // 30th annual INCOSE international symposium, Cape Town, South Africa, July 18-23, 2020. San Francisco, CA: John Wiley & Sons. ISSN 2334-5837. 2020, vol. 30, iss. 1, p. 391-405. DOI: 10.1002/j.2334-5837.2020.00729.x.
3. Bankauskaite, Jovita. Comparative analysis of enterprise architecture frameworks // CEUR workshop proceedings: IVUS 2019 international conference on information technologies: proceedings of the international conference on information technologies, Kaunas, Lithuania, April 25, 2019 / edited by: Robertas Damasevicius, Tomas Krilavicius, Audrius Lopata, Dawid Połap, Marcin Woźniak. Aachen: CEUR-WS. ISSN 1613-0073. 2019, vol. 2470, p. 61-64.

CURRICULUM VITAE

Education:

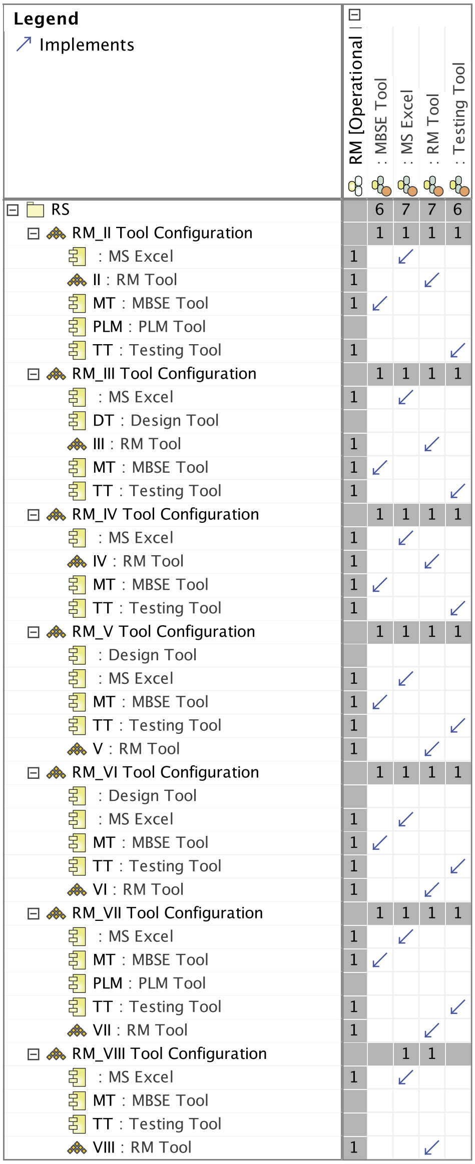
2018-2022	Doctoral studies in computer science engineering at Kaunas University of Technology
2021	Attended study module at the Technical University of Berlin (System Dynamics Modeling for Business Analysis)
2016-2018	Master degree in computer science at Kaunas University of Technology
2011-2016	Bachelor degree in computer science and associate degree in business at Vilnius Tech

Professional Experience:

2018–currently	Systems Analyst at *Dassault Systèmes*, Ltd.
2016–2018	Analyst at *ProIT*, Ltd.
2013–2016	Analyst at *Bull Baltija*, Ltd.

APPENDIXES

Appendix A. Traceability links between critical interfaces and alternatives



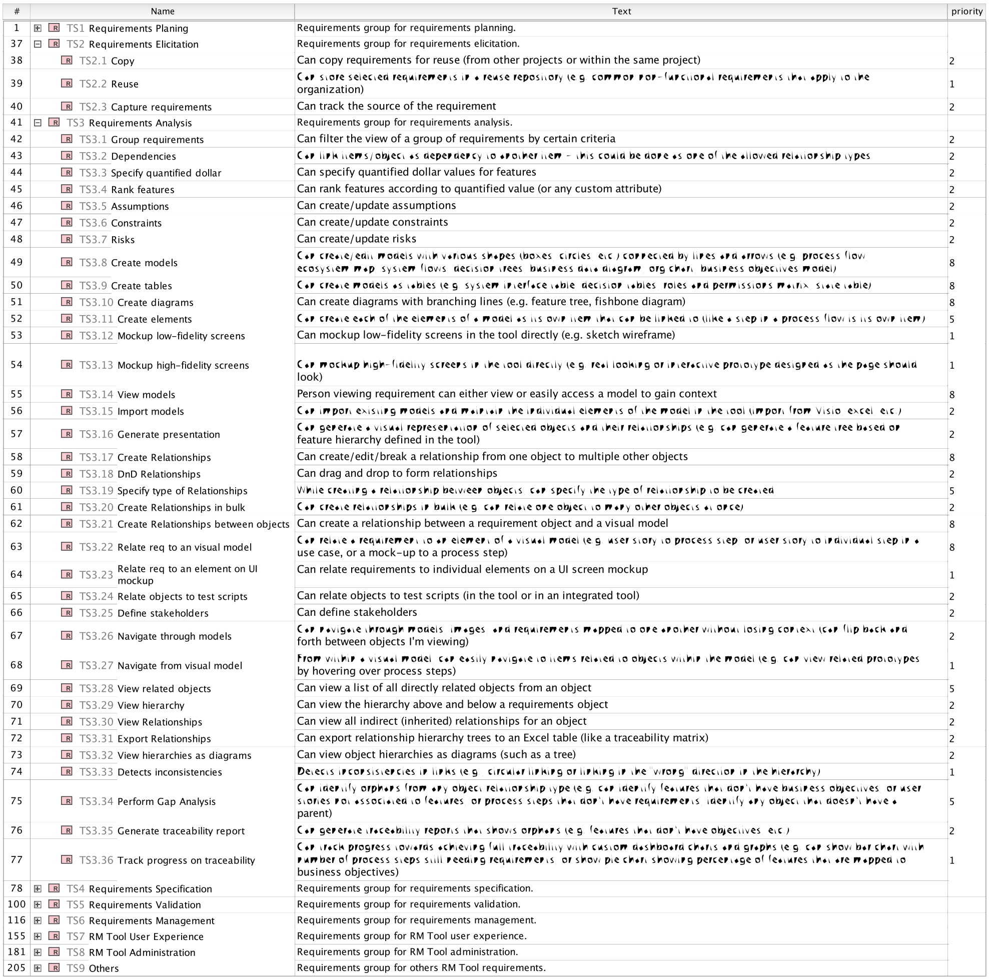
Legend

↗ Implements

	RM [Operational |	: MBSE Tool	: MS Excel	: RM Tool	: Testing Tool
RS		6	7	7	6
RM_II Tool Configuration		1	1	1	1
: MS Excel	1		↙		
II : RM Tool	1			↙	
MT : MBSE Tool	1	↙			
PLM : PLM Tool					
TT : Testing Tool	1				↙
RM_III Tool Configuration		1	1	1	1
: MS Excel	1		↙		
DT : Design Tool					
III : RM Tool	1			↙	
MT : MBSE Tool	1	↙			
TT : Testing Tool	1				↙
RM_IV Tool Configuration		1	1	1	1
: MS Excel	1		↙		
IV : RM Tool	1			↙	
MT : MBSE Tool	1	↙			
TT : Testing Tool	1				↙
RM_V Tool Configuration		1	1	1	1
: Design Tool					
: MS Excel	1		↙		
MT : MBSE Tool	1	↙			
TT : Testing Tool	1				↙
V : RM Tool	1			↙	
RM_VI Tool Configuration		1	1	1	1
: Design Tool					
: MS Excel	1		↙		
MT : MBSE Tool	1	↙			
TT : Testing Tool	1				↙
VI : RM Tool	1			↙	
RM_VII Tool Configuration		1	1	1	1
: MS Excel	1		↙		
MT : MBSE Tool	1	↙			
PLM : PLM Tool					
TT : Testing Tool	1				↙
VII : RM Tool	1			↙	
RM_VIII Tool Configuration			1	1	
: MS Excel	1		↙		
MT : MBSE Tool					
TT : Testing Tool					
VIII : RM Tool	1			↙	

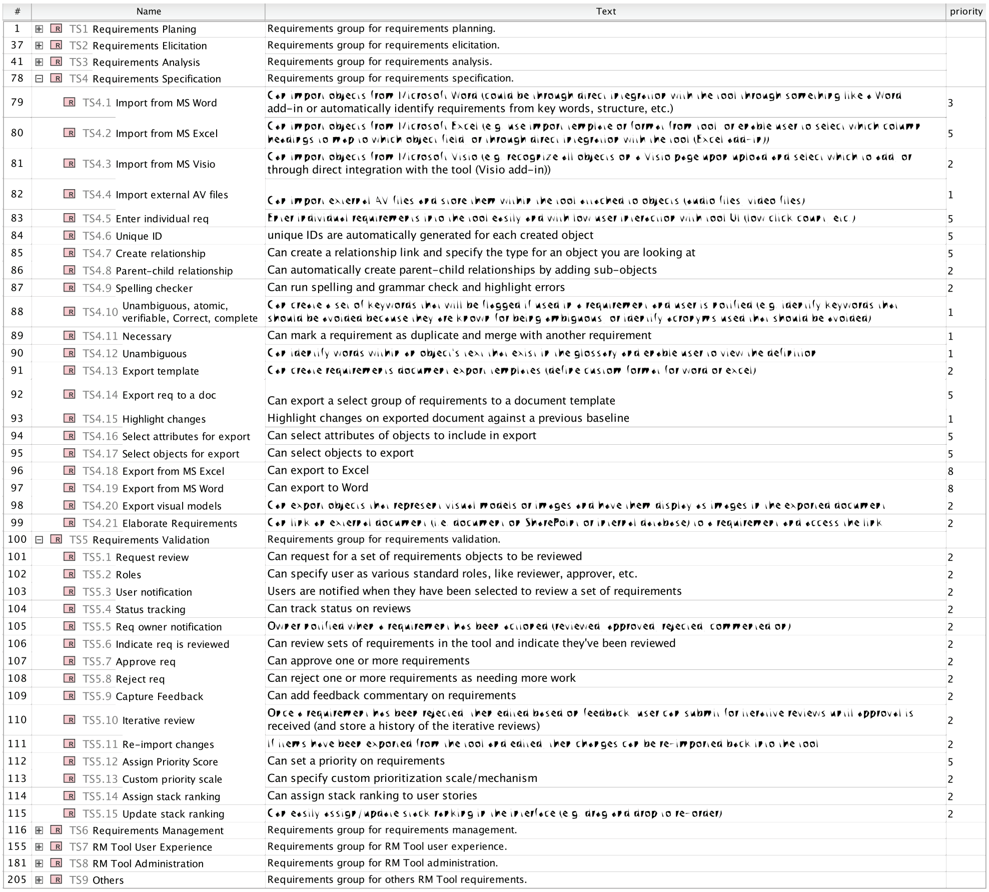
Fig. A.1. Traceability links between critical interfaces and alternatives

Appendix B. Selection criteria for the trade study of RM tool



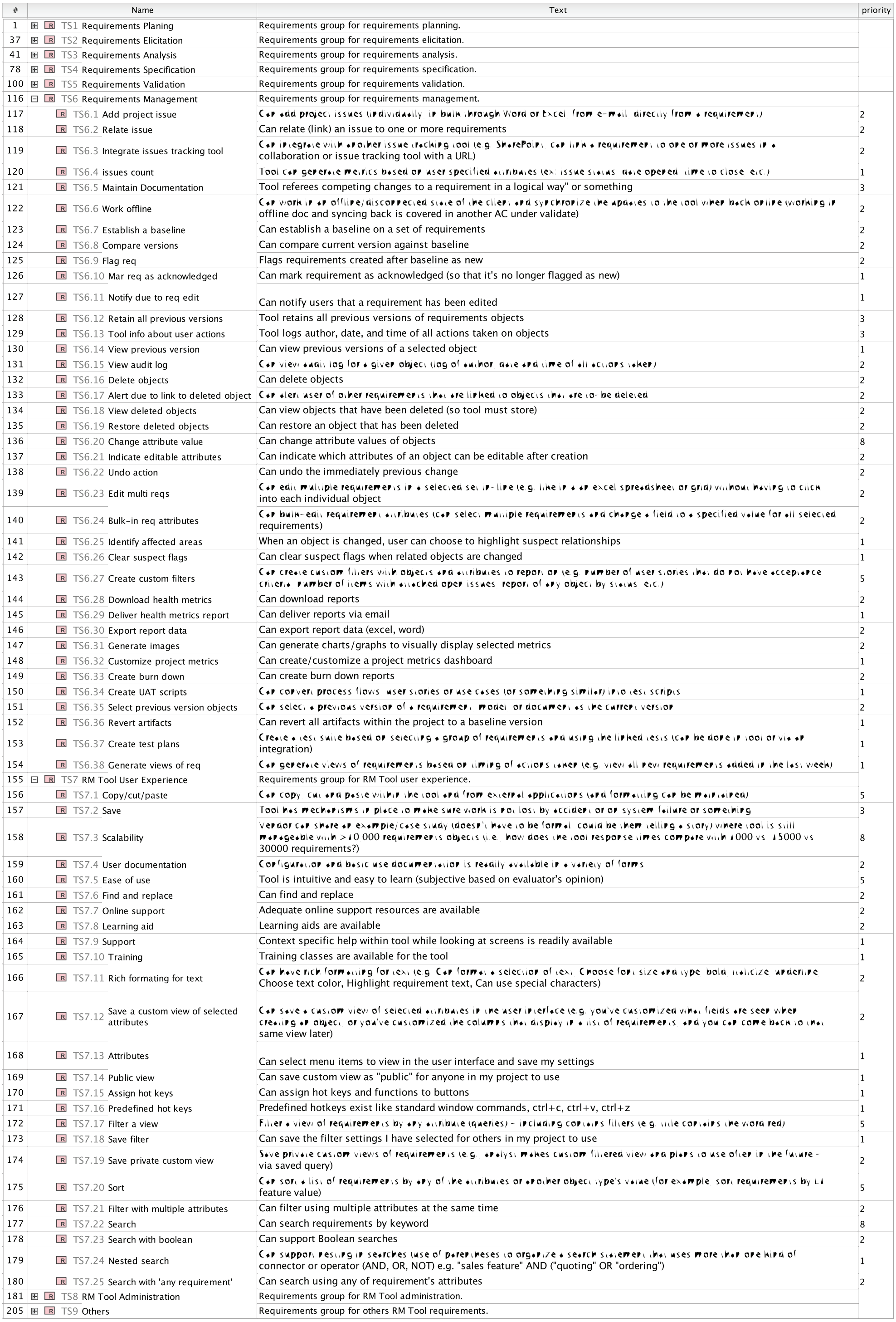
#	Name	Text	priority
1	⊞ R TS1 Requirements Planing	Requirements group for requirements planning.	
37	⊟ R TS2 Requirements Elicitation	Requirements group for requirements elicitation.	
38	R TS2.1 Copy	Can copy requirements for reuse (from other projects or within the same project)	2
39	R TS2.2 Reuse	Can store selected requirements in a reuse repository (e.g. common non-functional requirements that apply to the organization)	1
40	R TS2.3 Capture requirements	Can track the source of the requirement	2
41	⊟ R TS3 Requirements Analysis	Requirements group for requirements analysis.	
42	R TS3.1 Group requirements	Can filter the view of a group of requirements by certain criteria	2
43	R TS3.2 Dependencies	Can link items/object as dependency to another item - this could be done as one of the allowed relationship types	2
44	R TS3.3 Specify quantified dollar	Can specify quantified dollar values for features	2
45	R TS3.4 Rank features	Can rank features according to quantified value (or any custom attribute)	2
46	R TS3.5 Assumptions	Can create/update assumptions	2
47	R TS3.6 Constraints	Can create/update constraints	2
48	R TS3.7 Risks	Can create/update risks	2
49	R TS3.8 Create models	Can create/edit models with various shapes (boxes, circles, etc.) connected by lines and arrows (e.g. process flow, ecosystem map, system flows, decision trees, business data diagram, org chart, business objectives model)	8
50	R TS3.9 Create tables	Can create models as tables (e.g. system interface table, decision tables, roles and permissions matrix, state table)	8
51	R TS3.10 Create diagrams	Can create diagrams with branching lines (e.g. feature tree, fishbone diagram)	8
52	R TS3.11 Create elements	Can create each of the elements of a model as its own item that can be linked to (like a step in a process flow is its own item)	5
53	R TS3.12 Mockup low-fidelity screens	Can mockup low-fidelity screens in the tool directly (e.g. sketch wireframe)	1
54	R TS3.13 Mockup high-fidelity screens	Can mockup high-fidelity screens in the tool directly (e.g. real looking or interactive prototype designed as the page should look)	1
55	R TS3.14 View models	Person viewing requirement can either view or easily access a model to gain context	8
56	R TS3.15 Import models	Can import existing models and maintain the individual elements of the model in the tool (import from Visio, excel, etc.)	2
57	R TS3.16 Generate presentation	Can generate a visual representation of selected objects and their relationships (e.g. can generate a feature tree based on feature hierarchy defined in the tool)	2
58	R TS3.17 Create Relationships	Can create/edit/break a relationship from one object to multiple other objects	8
59	R TS3.18 DnD Relationships	Can drag and drop to form relationships	2
60	R TS3.19 Specify type of Relationships	While creating a relationship between objects, can specify the type of relationship to be created	5
61	R TS3.20 Create Relationships in bulk	Can create relationships in bulk (e.g. can relate one object to many other objects at once)	2
62	R TS3.21 Create Relationships between objects	Can create a relationship between a requirement object and a visual model	8
63	R TS3.22 Relate req to an visual model	Can relate a requirement to an element of a visual model (e.g. user story to process step, or user story to individual step in a use case, or a mock-up to a process step)	8
64	R TS3.23 Relate req to an element on UI mockup	Can relate requirements to individual elements on a UI screen mockup	1
65	R TS3.24 Relate objects to test scripts	Can relate objects to test scripts (in the tool or in an integrated tool)	2
66	R TS3.25 Define stakeholders	Can define stakeholders	2
67	R TS3.26 Navigate through models	Can navigate through models, images, and requirements mapped to one another without losing context (can flip back and forth between objects I'm viewing)	2
68	R TS3.27 Navigate from visual model	From within a visual model, can easily navigate to items related to objects within the model (e.g. can view related prototypes by hovering over process steps)	1
69	R TS3.28 View related objects	Can view a list of all directly related objects from an object	5
70	R TS3.29 View hierarchy	Can view the hierarchy above and below a requirements object	2
71	R TS3.30 View Relationships	Can view all indirect (inherited) relationships for an object	2
72	R TS3.31 Export Relationships	Can export relationship hierarchy trees to an Excel table (like a traceability matrix)	2
73	R TS3.32 View hierarchies as diagrams	Can view object hierarchies as diagrams (such as a tree)	2
74	R TS3.33 Detects inconsistencies	Detects inconsistencies in links (e.g., circular linking or linking in the "wrong" direction in the hierarchy)	1
75	R TS3.34 Perform Gap Analysis	Can identify orphans from any object relationship type (e.g. can identify features that don't have business objectives, or user stories not associated to features, or process steps that don't have requirements, identify any object that doesn't have a parent)	5
76	R TS3.35 Generate traceability report	Can generate traceability reports that shows orphans (e.g. features that don't have objectives, etc.)	2
77	R TS3.36 Track progress on traceability	Can track progress towards achieving full traceability with custom dashboard charts and graphs (e.g. can show bar chart with number of process steps still needing requirements, or show pie chart showing percentage of features that are mapped to business objectives)	1
78	⊞ R TS4 Requirements Specification	Requirements group for requirements specification.	
100	⊞ R TS5 Requirements Validation	Requirements group for requirements validation.	
116	⊞ R TS6 Requirements Management	Requirements group for requirements management.	
155	⊞ R TS7 RM Tool User Experience	Requirements group for RM Tool user experience.	
181	⊞ R TS8 RM Tool Administration	Requirements group for RM Tool administration.	
205	⊞ R TS9 Others	Requirements group for others RM Tool requirements.	

Fig. B.2. Selection criteria of requirements elicitation and analysis groups



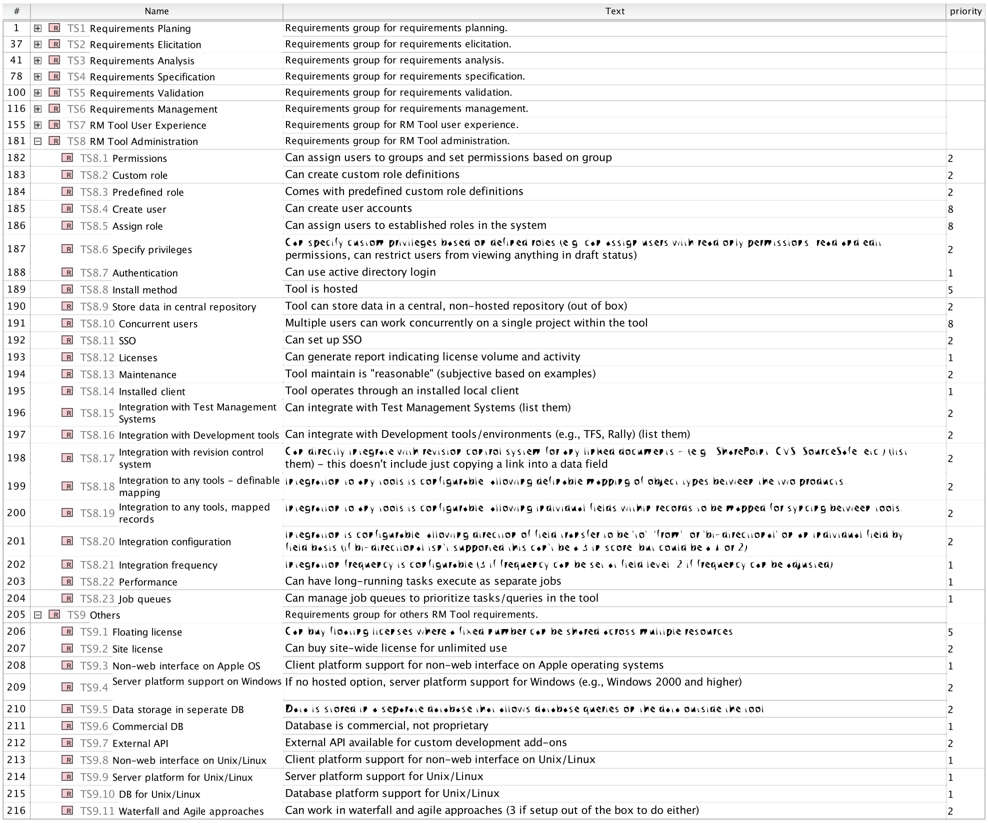
#	Name	Text	priority
1	TS1 Requirements Planing	Requirements group for requirements planning.	
37	TS2 Requirements Elicitation	Requirements group for requirements elicitation.	
41	TS3 Requirements Analysis	Requirements group for requirements analysis.	
78	TS4 Requirements Specification	Requirements group for requirements specification.	
79	TS4.1 Import from MS Word	Can import objects from Microsoft Word (could be through direct integration with the tool through something like a Word add-in or automatically identify requirements from key words, structure, etc.)	3
80	TS4.2 Import from MS Excel	Can import objects from Microsoft Excel (e.g. use import template or format from tool, or enable user to select which column headings to map to which object field, or through direct integration with the tool (Excel add-in))	5
81	TS4.3 Import from MS Visio	Can import objects from Microsoft Visio (e.g. recognize all objects on a Visio page upon upload and select which to add, or through direct integration with the tool (Visio add-in))	2
82	TS4.4 Import external AV files	Can import external AV files and store them within the tool attached to objects (audio files, video files)	1
83	TS4.5 Enter individual req	Enter individual requirements into the tool easily and with low user interaction with tool UI (low click count, etc.)	5
84	TS4.6 Unique ID	unique IDs are automatically generated for each created object	5
85	TS4.7 Create relationship	Can create a relationship link and specify the type for an object you are looking at	5
86	TS4.8 Parent-child relationship	Can automatically create parent-child relationships by adding sub-objects	2
87	TS4.9 Spelling checker	Can run spelling and grammar check and highlight errors	2
88	TS4.10 Unambiguous, atomic, verifiable, Correct, complete	Can create a set of keywords that will be flagged if used in a requirement and user is notified (e.g. identify keywords that should be avoided because they are known for being ambiguous, or identify acronyms used that should be avoided)	1
89	TS4.11 Necessary	Can mark a requirement as duplicate and merge with another requirement	1
90	TS4.12 Unambiguous	Can identify words within an object's text that exist in the glossary and enable user to view the definition	1
91	TS4.13 Export template	Can create requirements document export templates (define custom format for word or excel)	2
92	TS4.14 Export req to a doc	Can export a select group of requirements to a document template	5
93	TS4.15 Highlight changes	Highlight changes on exported document against a previous baseline	1
94	TS4.16 Select attributes for export	Can select attributes of objects to include in export	5
95	TS4.17 Select objects for export	Can select objects to export	5
96	TS4.18 Export from MS Excel	Can export to Excel	8
97	TS4.19 Export from MS Word	Can export to Word	8
98	TS4.20 Export visual models	Can export objects that represent visual models or images and have them display as images in the exported document	2
99	TS4.21 Elaborate Requirements	Can link an external document (i.e. document on SharePoint or internal database) to a requirement and access the link	2
100	TS5 Requirements Validation	Requirements group for requirements validation.	
101	TS5.1 Request review	Can request for a set of requirements objects to be reviewed	2
102	TS5.2 Roles	Can specify user as various standard roles, like reviewer, approver, etc.	2
103	TS5.3 User notification	Users are notified when they have been selected to review a set of requirements	2
104	TS5.4 Status tracking	Can track status on reviews	2
105	TS5.5 Req owner notification	Owner notified when a requirement has been actioned (reviewed, approved, rejected, commented on)	2
106	TS5.6 Indicate req is reviewed	Can review sets of requirements in the tool and indicate they've been reviewed	2
107	TS5.7 Approve req	Can approve one or more requirements	2
108	TS5.8 Reject req	Can reject one or more requirements as needing more work	2
109	TS5.9 Capture Feedback	Can add feedback commentary on requirements	2
110	TS5.10 Iterative review	Once a requirement has been rejected, then edited based on feedback, user can submit for iterative reviews until approval is received (and store a history of the iterative reviews)	2
111	TS5.11 Re-import changes	If items have been exported from the tool and edited, then changes can be re-imported back into the tool	2
112	TS5.12 Assign Priority Score	Can set a priority on requirements	5
113	TS5.13 Custom priority scale	Can specify custom prioritization scale/mechanism	2
114	TS5.14 Assign stack ranking	Can assign stack ranking to user stories	2
115	TS5.15 Update stack ranking	Can easily assign/update stack ranking in the interface (e.g. drag and drop to re-order)	2
116	TS6 Requirements Management	Requirements group for requirements management.	
155	TS7 RM Tool User Experience	Requirements group for RM Tool user experience.	
181	TS8 RM Tool Administration	Requirements group for RM Tool administration.	
205	TS9 Others	Requirements group for others RM Tool requirements.	

Fig. B.3. Selection criteria of requirements specification and validation groups



#	Name	Text	priority
1	⊞ TS1 Requirements Planing	Requirements group for requirements planning.	
37	⊞ TS2 Requirements Elicitation	Requirements group for requirements elicitation.	
41	⊞ TS3 Requirements Analysis	Requirements group for requirements analysis.	
78	⊞ TS4 Requirements Specification	Requirements group for requirements specification.	
100	⊞ TS5 Requirements Validation	Requirements group for requirements validation.	
116	⊟ TS6 Requirements Management	Requirements group for requirements management.	
117	TS6.1 Add project issue	Can add project issues (individually, in bulk through Word or Excel, from e-mail, directly from a requirement)	2
118	TS6.2 Relate issue	Can relate (link) an issue to one or more requirements	2
119	TS6.3 Integrate issues tracking tool	Can integrate with another issue tracking tool (e.g. SharePoint, can link a requirement to one or more issues in a collaboration or issue tracking tool with a URL)	2
120	TS6.4 issues count	Tool can generate metrics based on user specified attributes (ex: issue status, date opened, time to close, etc.)	1
121	TS6.5 Maintain Documentation	Tool referees competing changes to a requirement in a logical way" or something	3
122	TS6.6 Work offline	Can work in an offline/disconnected state of the client and synchronize the updates to the tool when back online (working in offline doc and syncing back is covered in another AC under validate)	2
123	TS6.7 Establish a baseline	Can establish a baseline on a set of requirements	2
124	TS6.8 Compare versions	Can compare current version against baseline	2
125	TS6.9 Flag req	Flags requirements created after baseline as new	2
126	TS6.10 Mar req as acknowledged	Can mark requirement as acknowledged (so that it's no longer flagged as new)	1
127	TS6.11 Notify due to req edit	Can notify users that a requirement has been edited	1
128	TS6.12 Retain all previous versions	Tool retains all previous versions of requirements objects	3
129	TS6.13 Tool info about user actions	Tool logs author, date, and time of all actions taken on objects	3
130	TS6.14 View previous version	Can view previous versions of a selected object	1
131	TS6.15 View audit log	Can view audit log for a given object (log of author, date and time of all actions taken)	2
132	TS6.16 Delete objects	Can delete objects	2
133	TS6.17 Alert due to link to deleted object	Can alert user of other requirements that are linked to objects that are to-be deleted	2
134	TS6.18 View deleted objects	Can view objects that have been deleted (so tool must store)	2
135	TS6.19 Restore deleted objects	Can restore an object that has been deleted	2
136	TS6.20 Change attribute value	Can change attribute values of objects	8
137	TS6.21 Indicate editable attributes	Can indicate which attributes of an object can be editable after creation	2
138	TS6.22 Undo action	Can undo the immediately previous change	2
139	TS6.23 Edit multi reqs	Can edit multiple requirements in a selected set in-line (e.g. like in a an excel spreadsheet or grid) without having to click into each individual object	2
140	TS6.24 Bulk-in req attributes	Can bulk-edit requirement attributes (can select multiple requirements and change a field to a specified value for all selected requirements)	2
141	TS6.25 Identify affected areas	When an object is changed, user can choose to highlight suspect relationships	1
142	TS6.26 Clear suspect flags	Can clear suspect flags when related objects are changed	1
143	TS6.27 Create custom filters	Can create custom filters with objects and attributes to report on (e.g. number of user stories that do not have acceptance criteria, number of items with attached open issues, report of any object by status, etc.)	5
144	TS6.28 Download health metrics	Can download reports	2
145	TS6.29 Deliver health metrics report	Can deliver reports via email	1
146	TS6.30 Export report data	Can export report data (excel, word)	2
147	TS6.31 Generate images	Can generate charts/graphs to visually display selected metrics	2
148	TS6.32 Customize project metrics	Can create/customize a project metrics dashboard	1
149	TS6.33 Create burn down	Can create burn down reports	2
150	TS6.34 Create UAT scripts	Can convert process flows, user stories or use cases (or something similar) into test scripts	1
151	TS6.35 Select previous version objects	Can select a previous version of a requirement, model, or document as the current version	2
152	TS6.36 Revert artifacts	Can revert all artifacts within the project to a baseline version	1
153	TS6.37 Create test plans	Create a test suite based on selecting a group of requirements and using the linked tests (can be done in tool or via an integration)	1
154	TS6.38 Generate views of req	Can generate views of requirements based on timing of actions taken (e.g. view all new requirements added in the last week)	1
155	⊟ TS7 RM Tool User Experience	Requirements group for RM Tool user experience.	
156	TS7.1 Copy/cut/paste	Can copy, cut and paste within the tool and from external applications (and formatting can be maintained)	5
157	TS7.2 Save	Tool has mechanisms in place to make sure work is not lost by accident or on system failure or something	3
158	TS7.3 Scalability	Vendor can share an example/case study (doesn't have to be formal, could be them telling a story) where tool is still manageable with >10,000 requirements objects (i.e., how does the tool response times compare with 1000 vs. 15000 vs. 30000 requirements?)	8
159	TS7.4 User documentation	Configuration and basic use documentation is readily available in a variety of forms	2
160	TS7.5 Ease of use	Tool is intuitive and easy to learn (subjective based on evaluator's opinion)	5
161	TS7.6 Find and replace	Can find and replace	2
162	TS7.7 Online support	Adequate online support resources are available	2
163	TS7.8 Learning aid	Learning aids are available	2
164	TS7.9 Support	Context specific help within tool while looking at screens is readily available	1
165	TS7.10 Training	Training classes are available for the tool	1
166	TS7.11 Rich formating for text	Can have rich formatting for text (e.g. Can format a selection of text, Choose font size and type, bold, italicize, underline, Choose text color, Highlight requirement text, Can use special characters)	2
167	TS7.12 Save a custom view of selected attributes	Can save a custom view of selected attributes in the user interface (e.g. you've customized what fields are seen when creating an object, or you've customized the columns that display in a list of requirements, and you can come back to that same view later)	2
168	TS7.13 Attributes	Can select menu items to view in the user interface and save my settings	1
169	TS7.14 Public view	Can save custom view as "public" for anyone in my project to use	1
170	TS7.15 Assign hot keys	Can assign hot keys and functions to buttons	1
171	TS7.16 Predefined hot keys	Predefined hotkeys exist like standard window commands, ctrl+c, ctrl+v, ctrl+z	1
172	TS7.17 Filter a view	Filter a view of requirements by any attribute (queries) - including contains filters (e.g. title contains the word red)	5
173	TS7.18 Save filter	Can save the filter settings I have selected for others in my project to use	1
174	TS7.19 Save private custom view	Save private custom views of requirements (e.g., analyst makes custom filtered view and plans to use often in the future - via saved query)	2
175	TS7.20 Sort	Can sort a list of requirements by any of the attributes or another object type's value (for example, sort requirements by LA feature value)	5
176	TS7.21 Filter with multiple attributes	Can filter using multiple attributes at the same time	2
177	TS7.22 Search	Can search requirements by keyword	8
178	TS7.23 Search with boolean	Can support Boolean searches	2
179	TS7.24 Nested search	Can support nesting in searches (use of parentheses to organize a search statement that uses more than one kind of connector or operator (AND, OR, NOT) e.g. "sales feature" AND ("quoting" OR "ordering")	1
180	TS7.25 Search with 'any requirement'	Can search using any of requirement's attributes	2
181	⊞ TS8 RM Tool Administration	Requirements group for RM Tool administration.	
205	⊞ TS9 Others	Requirements group for others RM Tool requirements.	

Fig. B.4. Selection criteria of requirements management and RM tool user experience groups



#	Name	Text	priority
1	⊞ TS1 Requirements Planing	Requirements group for requirements planning.	
37	⊞ TS2 Requirements Elicitation	Requirements group for requirements elicitation.	
41	⊞ TS3 Requirements Analysis	Requirements group for requirements analysis.	
78	⊞ TS4 Requirements Specification	Requirements group for requirements specification.	
100	⊞ TS5 Requirements Validation	Requirements group for requirements validation.	
116	⊞ TS6 Requirements Management	Requirements group for requirements management.	
155	⊞ TS7 RM Tool User Experience	Requirements group for RM Tool user experience.	
181	⊟ TS8 RM Tool Administration	Requirements group for RM Tool administration.	
182	TS8.1 Permissions	Can assign users to groups and set permissions based on group	2
183	TS8.2 Custom role	Can create custom role definitions	2
184	TS8.3 Predefined role	Comes with predefined custom role definitions	2
185	TS8.4 Create user	Can create user accounts	8
186	TS8.5 Assign role	Can assign users to established roles in the system	8
187	TS8.6 Specify privileges	Can specify custom privileges based on defined roles (e.g. can assign users with read only permissions, read and edit permissions, can restrict users from viewing anything in draft status)	2
188	TS8.7 Authentication	Can use active directory login	1
189	TS8.8 Install method	Tool is hosted	5
190	TS8.9 Store data in central repository	Tool can store data in a central, non-hosted repository (out of box)	2
191	TS8.10 Concurrent users	Multiple users can work concurrently on a single project within the tool	8
192	TS8.11 SSO	Can set up SSO	2
193	TS8.12 Licenses	Can generate report indicating license volume and activity	1
194	TS8.13 Maintenance	Tool maintain is "reasonable" (subjective based on examples)	2
195	TS8.14 Installed client	Tool operates through an installed local client	1
196	TS8.15 Integration with Test Management Systems	Can integrate with Test Management Systems (list them)	2
197	TS8.16 Integration with Development tools	Can integrate with Development tools/environments (e.g., TFS, Rally) (list them)	2
198	TS8.17 Integration with revision control system	Can directly integrate with revision control system for any linked documents - (e.g., SharePoint, CVS, SourceSafe, etc.) (list them) - this doesn't include just copying a link into a data field	2
199	TS8.18 Integration to any tools - definable mapping	Integration to any tools is configurable, allowing definable mapping of object types between the two products.	2
200	TS8.19 Integration to any tools, mapped records	Integration to any tools is configurable, allowing individual fields within records to be mapped for syncing between tools.	2
201	TS8.20 Integration configuration	Integration is configurable, allowing direction of field transfer to be 'to', 'from', or 'bi-directional' on an individual field by field basis (if bi-directional isn't supported this can't be a 3 in score, but could be a 1 or 2)	2
202	TS8.21 Integration frequency	Integration frequency is configurable (3 if frequency can be set at field level, 2 if frequency can be adjusted)	1
203	TS8.22 Performance	Can have long-running tasks execute as separate jobs	1
204	TS8.23 Job queues	Can manage job queues to prioritize tasks/queries in the tool	1
205	⊟ TS9 Others	Requirements group for others RM Tool requirements.	
206	TS9.1 Floating license	Can buy floating licenses where a fixed number can be shared across multiple resources	5
207	TS9.2 Site license	Can buy site-wide license for unlimited use	2
208	TS9.3 Non-web interface on Apple OS	Client platform support for non-web interface on Apple operating systems	1
209	TS9.4 Server platform support on Windows	If no hosted option, server platform support for Windows (e.g., Windows 2000 and higher)	2
210	TS9.5 Data storage in seperate DB	Data is stored in a separate database that allows database queries on the data outside the tool	2
211	TS9.6 Commercial DB	Database is commercial, not proprietary	1
212	TS9.7 External API	External API available for custom development add-ons	2
213	TS9.8 Non-web interface on Unix/Linux	Client platform support for non-web interface on Unix/Linux	1
214	TS9.9 Server platform for Unix/Linux	Server platform support for Unix/Linux	1
215	TS9.10 DB for Unix/Linux	Database platform support for Unix/Linux	1
216	TS9.11 Waterfall and Agile approaches	Can work in waterfall and agile approaches (3 if setup out of the box to do either)	2

Fig. B.5. Selection criteria of RM tool administration and others groups

Appendix C. Alternative RM tool with determined scores by selection criteria

Table C.1. Alternative RM tool with determined scores by selection criteria

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
1	**TS1 Requirements Planning**							
2	TS1.1 Input project knows	1	1	0	0	0	0	0
3	TS1.2 Estimate requirements work	3	2	3	1	0	2	2
4	TS1.3 Link tasks	3	2	3	1	2	3	3
5	TS1.4 Input tasks with estimates	3	2	3	1	2	2	2
6	TS1.5 Graph progress	1	0	3	1	1	3	3
7	TS1.6 Create requirements plan	3	2	3	1	2	3	3
8	TS1.7 Manage project portfolio	3	2	3	3	3	1	1
9	TS1.8 Customize methodology checklist	1	1	2	1	3	1	1
10	TS1.9 Import Org Chart	3	0	2	3	2	1	1
11	TS1.10 Track participants	3	3	3	3	3	2	2
12	TS1.11 Create personas	1	2	1	3	3	1,5	1,5
13	TS1.12 Link personas	3	2	2	3	3	2	2
14	TS1.13 Define object hierarchy	2	3	2	3	3	2	2
15	TS1.14 Add custom object type	3	3	3	3	3	3	3
16	TS1.15 Req objects variation	3	3	3	3	3	2	2
17	TS1.16 Req architecture variation	3	3	3	3	3	2	2
18	TS1.17 Relationships constraints	3	3	1	3	3	3	3
19	TS1.18 Relationships support	3	3	2	3	3	3	3
20	TS1.19 Bidirectional relationships	3	3	2	3	3	3	3
21	TS1.20 Visual layout of the hierarchy relationship	2	1	1	3	3	3	3
22	TS1.21 Req architecture templates	2	3	2	3	2	2	2
23	TS1.22 Create req architecture templates	1,5	0	2	3	3	3	3
24	TS1.23 Store relationship template	2	0	1	3	3	3	3
25	TS1.24 Fields customization	1,5	2	1	3	3	3	3
26	TS1.25 Define object attributes	1	0	2	3	2	3	3
27	TS1.26 Define field mandatory	1	0	3	3	3	3	3
28	TS1.27 Define field format	1	3	1	3	2	3	3
29	TS1.28 Define calculated fields	1	1	1	1	2	3	3
30	TS1.29 Customize tools	3	0	3	2	3	3	3
31	TS1.30 Create Glossary	1	1	1	1,5	1	2	2
32	TS1.31 Upload Glossary	0	2	1	1,5	1	1	1
33	TS1.32 Assign owner of req	3	1	2	3	3	3	3

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
34	TS1.33 Receive notification	3	0	3	3	3	2	2
35	TS1.34 Create requirements plan	1,5	2	3	3	3	2	2
36	TS1.35 Visually see features	1,5	2	3	3	3	2	2
37	**TS2 Requirements Elicitation**							
38	TS2.1 Copy	1	1	3	1,5	3	3	1
39	TS2.2 Reuse	1	0	2	1,5	3	2	1
40	TS2.3 Capture requirements	3	3	3	3	3	3	3
41	**TS3 Requirements Analysis**							
42	TS3.1 Group requirements	0	0	3	3	3	3	3
43	TS3.2 Dependencies	3	3	3	3	3	3	3
44	TS3.3 Specify quantified dollar	1	1	3	3	3	2	3
45	TS3.4 Rank features	0	1	3	2	2,5	1	1
46	TS3.5 Assumptions	1	3	2	3	3	2	2
47	TS3.6 Constraints	1	3	2	3	3	2	2
48	TS3.7 Risks	1	3	2	3	3	2	2
49	TS3.8 Create models	3	2	0	1	3	2	2
50	TS3.9 Create tables	2	2	1	3	2	2	2
51	TS3.10 Create diagrams	3	2	0	1	3	2	2
52	TS3.11 Create elements	0	0	1	1	1	2	2
53	TS3.12 Mockup low-fidelity screens	3	0	0	1	1	1,5	0
54	TS3.13 Mockup high-fidelity screens	1	0	0	1	0	1,5	0
55	TS3.14 View models	3	3	1	3	3	3	2
56	TS3.15 Import models	3	0	0	1	3	0	3
57	TS3.16 Generate presentation	1	2	0	3	2	3	2
58	TS3.17 Create Relationships	3	3	3	3	3	3	3
59	TS3.18 DnD Relationships	0	0	0	2	0	2	3
60	TS3.19 Specify type of Relationships	3	3	3	3	3	1	3
61	TS3.20 Create Relationships in bulk	3	0	2	3	2	3	3
62	TS3.21 Create Relationships between objects	2	2	1	3	3	1	3
63	TS3.22 Relate req to an visual model	0	0	0	1,5	1	2	2
64	TS3.23 Relate req to an element on UI mockup	0,5	0	0	1	0	1,5	0
65	TS3.24 Relate objects to test scripts	3	1	3	3	3	3	3
66	TS3.25 Define stakeholders	1	1	2	3	3	2	3
67	TS3.26 Navigate through models	3	1	1	3	3	2	2
68	TS3.27 Navigate from visual model	0	1	0	3	1	2	2

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
69	TS3.28 View related objects	3	2	3	3	3	3	3
70	TS3.29 View hierarchy	3	1	2	3	3	3	3
71	TS3.30 View Relationships	3	1	0	3	3	3	3
72	TS3.31 Export Relationships	3	2	1	3	3	3	2
73	TS3.32 View hierarchies as diagrams	0	1	0	3	2	3	2
74	TS3.33 Detects inconsistencies	1	0	0	3	2,5	3	1
75	TS3.34 Perform Gap Analysis	1	0	1	3	3	3	3
76	TS3.35 Generate traceability report	3	1	1	3	3	3	2
77	TS3.36 Track progress on traceability	1	2	1	3	2	3	3
78	**TS4 Requirements Specification**							
79	TS4.1 Import from MS Word	3	3	0	3	3	2	3
80	TS4.2 Import from MS Excel	3	3	3	3	3	2	3
81	TS4.3 Import from MS Visio	0	0	0	0	0	0	3
82	TS4.4 Import external AV files	3	3	2	3	3	0	2
83	TS4.5 Enter individual req	3	1	3	3	3	2	2
84	TS4.6 Unique ID	3	1	3	3	3	3	3
85	TS4.7 Create relationship	3	3	3	3	3	2	3
86	TS4.8 Parent-child relationship	3	0	3	3	3	2	3
87	TS4.9 Spelling checker	1	0	1	1	2	2	2
88	TS4.10 Unambiguous, atomic, verifiable, Correct, complete	0	0	0	3	2	0	3
89	TS4.11 Necessary	0	0	3	1,5	0	0	2
90	TS4.12 Unambiguous	0	0	0	1	0	0,5	0
91	TS4.13 Export template	2	3	1	3	2	3	3
92	TS4.14 Export req to a doc	3	3	1	3	3	3	3
93	TS4.15 Highlight changes	1	3	0	1,5	3	3	3
94	TS4.16 Select attributes for export	3	3	1	3	3	3	3
95	TS4.17 Select objects for export	3	3	2	3	3	3	3
96	TS4.18 Export from MS Excel	3	3	3	3	3	2	3
97	TS4.19 Export from MS Word	3	3	3	3	3	3	3
98	TS4.20 Export visual models	3	2	0	3	3	3	3
99	TS4.21 Elaborate Requirements	1	2	3	3	2,5	3	3
100	**TS5 Requirements Validation**							
101	TS5.1 Request review	3	2	3	1	3	3	3
102	TS5.2 Roles	3	0	3	3	3	3	3
103	TS5.3 User notification	3	2	3	3	3	2	3

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
104	TS5.4 Status tracking	3	1	3	3	3	3	3
105	TS5.5 Req owner notification	3	2	3	3	3	2	3
106	TS5.6 Indicate req is reviewed	3	0	3	3	3	3	3
107	TS5.7 Approve req	3	1	3	3	3	2	3
108	TS5.8 Reject req	3	1	3	3	3	2	3
109	TS5.9 Capture Feedback	3	3	3	3	3	3	3
110	TS5.10 Iterative review	2	1	3	3	3	2	3
111	TS5.11 Re-import changes	3	3	0	2	2	2	3
112	TS5.12 Assign Priority Score	3	1	3	3	3	3	3
113	TS5.13 Custom priority scale	1	0	3	3	3	2	3
114	TS5.14 Assign stack ranking	1	0	1	1,5	3	1	0
115	TS5.15 Update stack ranking	3	0	1	1,5	3	0	0
116	**TS6 Requirements Management**							
117	TS6.1 Add project issue	2	2	3	3	3	1	3
118	TS6.2 Relate issue	3	2	3	3	3	2	3
119	TS6.3 Integrate issues tracking tool	2	0	3	2,5	3	3	2
120	TS6.4 Issues count	0	2	3	3	1	3	2
121	TS6.5 Maintain Documentation	3	1	0	3	3	3	3
122	TS6.6 Work offline	1	1	0	1,5	1	3	1
123	TS6.7 Establish a baseline	3	2	2	2	3	3	3
124	TS6.8 Compare versions	1	3	1	2	3	2	3
125	TS6.9 Flag req	0,5	0	2	1	1	2,5	1
126	TS6.10 Mar req as acknowledged	0	0	1	2	1	1	1
127	TS6.11 Notify due to req edit	3	2	3	3	3	1	3
128	TS6.12 Retain all previous versions	3	3	3	3	3	3	3
129	TS6.13 Tool info about user actions	3	3	3	3	3	3	2
130	TS6.14 View previous version	3	3	0	3	3	3	3
131	TS6.15 View audit log	3	3	3	3	3	3	3
132	TS6.16 Delete objects	2	3	3	3	3	3	3
133	TS6.17 Alert due to link to deleted object	0	0	0	3	3	0	0
134	TS6.18 View deleted objects	0	0	0	2	3	3	2
135	TS6.19 Restore deleted objects	1	0	0	3	3	3	2
136	TS6.20 Change attribute value	1	2	3	3	3	3	2
137	TS6.21 Indicate editable attributes	1	0	2	3	3	3	3
138	TS6.22 Undo action	3	0	0	3	2	2	3

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
139	TS6.23 Edit multi reqs	3	1	0	3	3	2	1
140	TS6.24 Bulk-in req attributes	3	1	3	3	3	1	2
141	TS6.25 Identify affected areas	3	0	0	3	2	2	3
142	TS6.26 Clear suspect flags	3	0	0	3	2	2	3
143	TS6.27 Create custom filters	0	3	3	3	3	3	2
144	TS6.28 Download health metrics	1	2	3	3	3	3	3
145	TS6.29 Deliver health metrics report	1	0	3	1	3	2	1
146	TS6.30 Export report data	2	3	3	3	3	2	3
147	TS6.31 Generate images	1	1	3	3	2	3	3
148	TS6.32 Customize project metrics	1	3	3	3	3	1	3
149	TS6.33 Create burn down	1	0	3	1	1	2	1
150	TS6.34 Create UAT scripts	1	0	0	1	0	0	1
151	TS6.35 Select previous version objects	1	3	0	3	3	2	3
152	TS6.36 Revert artifacts	1	3	0	2	3	2	3
153	TS6.37 Create test plans	0	0	3	3	3	1	3
154	TS6.38 Generate views of req	1	1	3	3	2	2	3
155	**TS7 RM Tool User Experience**							
156	TS7.1 Copy/cut/paste	3	1	3	3	3	3	2
157	TS7.2 Save	1	1	0	3	1	0	1
158	TS7.3 Scalability	3	1	3	3	3	3	1
159	TS7.4 User documentation	3	1	3	3	3	2	3
160	TS7.5 Ease of use	1	2	2	2	2	1	2
161	TS7.6 Find and replace	1	0	0	2	2	2	0
162	TS7.7 Online support	2	3	3	3	3	2	3
163	TS7.8 Learning aid	3	1	3	3	3	1	3
164	TS7.9 Support	3	0	3	2	2	1	3
165	TS7.10 Training	3	3	3	3	3	2	3
166	TS7.11 Rich formatting for text	3	1	3	3	3	3	2
167	TS7.12 Save a custom view of selected attributes	1	0	3	3	1	3	2
168	TS7.13 Attributes	1	0	2	3	2	3	3
169	TS7.14 Public view	2	0	0	3	1	1	3
170	TS7.15 Assign hot keys	0	0	3	0	0	2	2
171	TS7.16 Predefined hot keys	3	1	3	3	3	3	2
172	TS7.17 Filter a view	0,5	3	3	3	2	3	3
173	TS7.18 Save filter	0,5	0	2	3	1	3	2

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
174	TS7.19 Save private custom view	0,5	0	0	3	3	3	3
175	TS7.20 Sort	3	1	3	2	3	3	3
176	TS7.21 Filter with multiple attributes	1	3	3	3	3	3	3
177	TS7.22 Search	3	3	3	3	3	3	3
178	TS7.23 Search with Boolean	1	3	3	3	3	3	1
179	TS7.24 Nested search	1	3	3	2	1	3	0
180	TS7.25 Search with 'any requirement'	1	2	3	3	3	3	3
181	**TS8 RM Tool Administration**							
182	TS8.1 Permissions	3	1	3	3	3	2	3
183	TS8.2 Custom role	3	0	3	3	3	2	3
184	TS8.3 Predefined role	3	1	3	3	3	3	3
185	TS8.4 Create user	2	3	3	3	3	3	3
186	TS8.5 Assign role	3	1	3	3	3	2	3
187	TS8.6 Specify privileges	3	1	3	3	3	2	3
188	TS8.7 Authentication	3	3	3	3	3	3	3
189	TS8.8 Install method	3	3	3	3	3	3	3
190	TS8.9 Store data in central repository	3	3	3	3	3	3	3
191	TS8.10 Concurrent users	3	3	3	3	3	3	3
192	TS8.11 SSO	3	3	3	3	3	3	3
193	TS8.12 Licenses	3	2	2	1	3	3	3
194	TS8.13 Maintenance	1	2	2	2	2	2	2
195	TS8.14 Installed client	0	0	3	3	3	3	3
196	TS8.15 Integration with Test Management Systems	2	0	3	3	2	3	0
197	TS8.16 Integration with Development tools	1	0		3	3	2	0
198	TS8.17 Integration with revision control system	2	0	3	2	0	2	0
199	TS8.18 Integration to any tools – definable mapping	3	0	3	3	3	3	0
200	TS8.19 Integration to any tools, mapped records	2	0	3	3	3	3	0
201	TS8.20 Integration configuration	1,5	0	3	3	3	3	0
202	TS8.21 Integration frequency	0	0	3	2	3	2	0
203	TS8.22 Performance	3	0	2	1	0	3	0
204	TS8.23 Job queues	0	0	1	0	3	1	0
205	**TS9 Others**							
206	TS9.1 Floating license	3	3	3	3	3	3	3
207	TS9.2 Site license	3	0	3	3	1	3	3

#	Criteria	RM I	RM II	RM III	RM IV	RM V	RM VI	RM VII
208	TS9.3 Non-web interface on Apple OS	3	0	0	0	3	3	0
209	TS9.4 Server platform support on Windows	3	3	3	3	3	3	3
210	TS9.5 Data storage in separate DB	3	3	3	3	3	3	3
211	TS9.6 Commercial DB	3	3	3	3	3	3	3
212	TS9.7 External API	3	3	3	3	3	3	3
213	TS9.8 Non-web interface on Unix/Linux	3	3	3	0	3	3	3
214	TS9.9 Server platform for Unix/Linux	3	3	3	0	3	3	3
215	TS9.10 DB for Unix/Linux	3	3	3	3	3	3	3
216	TS9.11 Waterfall and Agile approaches	2	2	2	2	3	3	2

Appendix D. Weighted score calculations of alternative RM tools

Table D.2. Weighted score calculations of alternative RM tools

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS1		0.0337										
TS1.1	1		0.5	0.0169	0	0	0	0	0	0	0.5	0.01685
TS1.2	1		0.3333	0.0112	0.3333	0.0112	0.1111	0.0037	0	0	0.2222	0.0075
TS1.3	1		0.25	0.0084	0.25	0.0084	0.0833	0.0028	0.1667	0.0056	0.25	0.0084
TS1.4	2		0.2727	0.0184	0.2727	0.0184	0.0909	0.0061	0.1818	0.0123	0.1818	0.0123
TS1.5	2		0.125	0.0084	0.375	0.0253	0.125	0.0084	0.125	0.0084	0.25	0.0169
TS1.6	1		0.2727	0.0092	0.2727	0.0092	0.0909	0.0031	0.1818	0.0061	0.1818	0.0061
TS1.7	1		0.2	0.0067	0.2	0.0067	0.2	0.0067	0.2	0.0067	0.2	0.0067
TS1.8	1		0.125	0.0042	0.25	0.0084	0.125	0.0042	0.375	0.0126	0.125	0.0042
TS1.9	2		0.2308	0.0156	0.1538	0.0104	0.2308	0.0156	0.1538	0.0104	0.2308	0.0156
TS1.10	1		0.2143	0.0072	0.2143	0.0072	0.2143	0.0072	0.2143	0.0072	0.1429	0.0048
TS1.11	2		0.0909	0.0061	0.0909	0.0061	0.2727	0.0184	0.2727	0.0184	0.2727	0.0184
TS1.12	2		0.2143	0.0144	0.1429	0.0096	0.2143	0.0144	0.2143	0.0144	0.2143	0.0144
TS1.13	8		0.1538	0.0415	0.1538	0.0415	0.2308	0.0622	0.2308	0.0622	0.2308	0.0622
TS1.14	8		0.2	0.0539	0.2	0.05392	0.2	0.05392	0.2	0.05392	0.2	0.05392
TS1.15	1		0.2	0.0067	0.2	0.00674	0.2	0.00674	0.2	0.00674	0.2	0.00674
TS1.16	2		0.2	0.0135	0.2	0.01348	0.2	0.01348	0.2	0.01348	0.2	0.01348
TS1.17	2		0.2308	0.0156	0.0769	0.0052	0.2308	0.0156	0.2308	0.0156	0.2308	0.0156
TS1.18	5		0.2143	0.0361	0.1429	0.0241	0.2143	0.0361	0.2143	0.0361	0.2143	0.0361
TS1.19	5		0.2143	0.0361	0.1429	0.0241	0.2143	0.0361	0.2143	0.0361	0.2143	0.0361
TS1.20	2		0.1818	0.0123	0.0909	0.0061	0.2727	0.0184	0.2727	0.0184	0.1818	0.0123
TS1.21	1		0.1818	0.0061	0.1818	0.0061	0.2727	0.0092	0.1818	0.0061	0.1818	0.0061

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS1.22	2		0.1304	0.0088	0.1739	0.0117	0.2609	0.0176	0.2609	0.0176	0.1739	0.0117
TS1.23	2		0.1818	0.0123	0.0909	0.0061	0.2727	0.0184	0.2727	0.0184	0.1818	0.0123
TS1.24	2		0.1429	0.0096	0.0952	0.0064	0.2857	0.0193	0.2857	0.0193	0.1905	0.0128
TS1.25	2		0.0909	0.0061	0.1818	0.0123	0.2727	0.0184	0.1818	0.0123	0.2727	0.0184
TS1.26	2		0.0769	0.0052	0.2308	0.0156	0.2308	0.0156	0.2308	0.0156	0.2308	0.0156
TS1.27	1		0.1	0.0034	0.1	0.0034	0.3	0.0101	0.2	0.0067	0.3	0.0101
TS1.28	1		0.125	0.0042	0.125	0.0042	0.125	0.0042	0.25	0.0084	0.375	0.0126
TS1.29	1		0.2308	0.0078	0.2308	0.0078	0.1538	0.0052	0.2308	0.0078	0.1538	0.0052
TS1.30	1		0.1333	0.0045	0.1333	0.0045	0.2	0.0067	0.1333	0.0045	0.4	0.0135
TS1.31	1		0	0.0000	0.1538	0.0052	0.2308	0.0078	0.1538	0.0052	0.4615	0.0156
TS1.32	3		0.2143	0.0217	0.1429	0.0144	0.2143	0.0217	0.2143	0.0217	0.2143	0.0217
TS1.33	1		0.25	0.0084	0.25	0.0084	0.25	0.0084	0.25	0.0084	0	0
TS1.34	2		0.12	0.0081	0.24	0.0162	0.24	0.0162	0.24	0.0162	0.16	0.0108
TS1.35	2		0.1111	0.0075	0.2222	0.0150	0.2222	0.0150	0.2222	0.0150	0.2222	0.0150
TS2		0.0201										
TS2.1	2		0.1053	0.0042	0.3158	0.0127	0.1579	0.0063	0.3158	0.0127	0.1053	0.0042
TS2.2	1		0.1176	0.0024	0.2353	0.0047	0.1765	0.0035	0.3529	0.0071	0.1176	0.0024
TS2.3	2		0.2	0.0080	0.2	0.0080	0.2	0.0080	0.2	0.0080	0.2	0.0080
TS3		0.1179										
TS3.1	2		0	0.0000	0.25	0.0590	0.25	0.0590	0.25	0.0590	0.25	0.0590
TS3.2	2		0.2	0.0472	0.2	0.0472	0.2	0.0472	0.2	0.0472	0.2	0.0472
TS3.3	2		0.0769	0.0181	0.2308	0.0544	0.2308	0.0544	0.2308	0.0544	0.2308	0.0544
TS3.4	2		0	0.0000	0.3529	0.0832	0.2353	0.0555	0.2941	0.0694	0.1176	0.0277
TS3.5	2		0.0909	0.0214	0.1818	0.0429	0.2727	0.0643	0.2727	0.0643	0.1818	0.0429
TS3.6	2		0.0909	0.0214	0.1818	0.0429	0.2727	0.0643	0.2727	0.0643	0.1818	0.0429

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS3.7	2		0.0909	0.0214	0.1818	0.0429	0.2727	0.0643	0.2727	0.0643	0.1818	0.0429
TS3.8	8		0.3333	0.3144	0	0	0.1111	0.1048	0.3333	0.3144	0.2222	0.2096
TS3.9	8		0.2	0.1886	0.1	0.0943	0.3	0.28296	0.2	0.18864	0.2	0.18864
TS3.10	8		0.3333	0.3144	0	0	0.1111	0.1048	0.3333	0.3144	0.2222	0.2096
TS3.11	5		0	0.0000	0.2	0.1179	0.2	0.1179	0.2	0.1179	0.4	0.2358
TS3.12	1	0.1179	0.6	0.0707	0	0	0.2	0.0236	0.2	0.0236	0	0
TS3.13	1		0.5	0.0590	0	0	0.5	0.0590	0	0	0	0
TS3.14	8		0.25	0.2358	0.0833	0.0786	0.25	0.2358	0.25	0.2358	0.1667	0.1572
TS3.15	2		0.3	0.0707	0	0	0.1	0.0236	0.3	0.0707	0.3	0.0707
TS3.16	2		0.125	0.0295	0	0	0.375	0.0884	0.25	0.0590	0.25	0.0590
TS3.17	8		0.2	0.1886	0.2	0.1886	0.2	0.1886	0.2	0.1886	0.2	0.1886
TS3.18	2		0	0.0000	0	0	0.4	0.0943	0	0	0.6	0.1415
TS3.19	5		0.2	0.1179	0.2	0.1179	0.2	0.1179	0.2	0.1179	0.2	0.1179
TS3.20	2		0.2308	0.0544	0.1538	0.0363	0.2308	0.0544	0.1538	0.0363	0.2308	0.0544
TS3.21	8		0.1667	0.1572	0.0833	0.0786	0.25	0.2358	0.25	0.2358	0.25	0.2358
TS3.22	8		0	0.0000	0	0	0.3333	0.3144	0.2222	0.2096	0.4444	0.4192
TS3.23	1		0.3333	0.0393	0	0	0.6667	0.0786	0	0	0	0
TS3.24	2		0.2	0.0472	0.2	0.0472	0.2	0.0472	0.2	0.0472	0.2	0.0472
TS3.25	2		0.0833	0.0197	0.1667	0.0393	0.25	0.0590	0.25	0.0590	0.25	0.0590
TS3.26	2		0.25	0.0590	0.0833	0.0197	0.25	0.0590	0.25	0.0590	0.1667	0.0393
TS3.27	1		0	0.0000	0	0	0.5	0.0590	0.1667	0.0197	0.3333	0.0393
TS3.28	5		0.2	0.1179	0.2	0.1179	0.2	0.1179	0.2	0.1179	0.2	0.1179
TS3.29	2		0.2143	0.0505	0.1429	0.0337	0.2143	0.0505	0.2143	0.0505	0.2143	0.0505
TS3.30	2		0.25	0.0590	0	0	0.25	0.0590	0.25	0.0590	0.25	0.0590

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS3.31	2		0.25	0.0590	0.0833	0.0197	0.25	0.0590	0.25	0.0590	0.1667	0.0393
TS3.32	2		0	0.0000	0	0	0.4286	0.1011	0.2857	0.0674	0.2857	0.0674
TS3.33	1		0.1333	0.0157	0	0	0.4	0.0472	0.3333	0.0393	0.1333	0.0157
TS3.34	5		0.0909	0.0536	0.0909	0.0536	0.2727	0.1608	0.2727	0.1608	0.2727	0.1608
TS3.35	2		0.25	0.0590	0.0833	0.0197	0.25	0.0590	0.25	0.0590	0.1667	0.0393
TS3.36	1		0.1	0.0118	0.1	0.0118	0.3	0.0354	0.2	0.0236	0.3	0.0354
TS4		0.0901										
TS4.1	3		0.25	0.0676	0	0	0.25	0.0676	0.25	0.0676	0.25	0.0676
TS4.2	5		0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901
TS4.3	2		0	0.0000	0	0	0	0	0	0	1	0.1802
TS4.4	1		0.2308	0.0208	0.1538	0.0139	0.2308	0.0208	0.2308	0.0208	0.1538	0.0139
TS4.5	5		0.2143	0.0965	0.2143	0.0965	0.2143	0.0965	0.2143	0.0965	0.1429	0.0644
TS4.6	5		0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901
TS4.7	5		0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901	0.2	0.0901
TS4.8	2		0.2	0.0360	0.2	0.0360	0.2	0.0360	0.2	0.0360	0.2	0.0360
TS4.9	2		0.1429	0.0257	0.1429	0.0257	0.1429	0.0257	0.2857	0.0515	0.2857	0.0515
TS4.10	1		0	0.0000	0	0	0.3750	0.0338	0.25	0.0225	0.375	0.0338
TS4.11	1		0	0.0000	0.4615	0.0416	0.2308	0.0208	0	0	0.3077	0.0277
TS4.12	1		0	0.0000	0	0	1	0.0901	0	0	0	0
TS4.13	2		0.1818	0.0328	0.0909	0.0164	0.2727	0.0491	0.1818	0.0328	0.2727	0.0491
TS4.14	5		0.2308	0.1040	0.0769	0.0347	0.2308	0.1040	0.2308	0.1040	0.2308	0.1040
TS4.15	1		0.1176	0.0106	0	0	0.1765	0.0159	0.3529	0.0318	0.3529	0.0318
TS4.16	5		0.2308	0.1040	0.0769	0.0347	0.2308	0.1040	0.2308	0.1040	0.2308	0.1040
TS4.17	5		0.2143	0.0965	0.1429	0.0644	0.2143	0.0965	0.2143	0.0965	0.2143	0.0965

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS4.18	8		0.2	0.1442	0.2	0.1442	0.2	0.1442	0.2	0.1442	0.2	0.1442
TS4.19	8		0.2	0.1442	0.2	0.1442	0.2	0.1442	0.2	0.1442	0.2	0.1442
TS4.20	2		0.25	0.0451	0	0	0.25	0.0451	0.25	0.0451	0.25	0.0451
TS4.21	2		0.08	0.0144	0.24	0.0432	0.24	0.0432	0.2	0.0360	0.24	0.0432
TS5		0.2082										
TS5.1	2		0.2308	0.0961	0.2308	0.0961	0.0769	0.0320	0.2308	0.0961	0.2308	0.0961
TS5.2	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.3	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.4	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.5	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.6	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.7	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.8	2		0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.9	2	0.2082	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833	0.2	0.0833
TS5.10	2		0.1429	0.0595	0.2143	0.0892	0.2143	0.0892	0.2143	0.0892	0.2143	0.0892
TS5.11	2		0.3	0.1249	0	0	0.2	0.0833	0.2	0.0833	0.3	0.1249
TS5.12	5		0.2	0.2082	0.2	0.2082	0.2	0.2082	0.2	0.2082	0.2	0.2082
TS5.13	2		0.0769	0.0320	0.2308	0.0961	0.2308	0.0961	0.2308	0.0961	0.2308	0.0961
TS5.14	2		0.1538	0.0641	0.1538	0.0641	0.2308	0.0961	0.4615	0.1922	0	0
TS5.15	2		0.3529	0.1470	0.1176	0.0490	0.1765	0.0735	0.3529	0.1470	0	0
TS6		0.3732										
TS6.1	2		0.1429	0.1066	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599
TS6.2	2		0.2	0.1493	0.2	0.1493	0.2	0.1493	0.2	0.1493	0.2	0.1493
TS6.3	2		0.16	0.1194	0.24	0.1791	0.2	0.14928	0.24	0.1791	0.16	0.1194
TS6.4	1		0	0.0000	0.3333	0.1244	0.3333	0.1244	0.1111	0.0415	0.2222	0.0829

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS6.5	3		0.25	0.2799	0	0	0.25	0.2799	0.25	0.2799	0.25	0.2799
TS6.6	2		0.2222	0.1659	0	0	0.3333	0.2488	0.2222	0.1659	0.2222	0.1659
TS6.7	2		0.2308	0.1722	0.1538	0.1148	0.1538	0.1148	0.2308	0.1722	0.2308	0.1722
TS6.8	2		0.1	0.0746	0.1	0.0746	0.2	0.1493	0.3	0.2239	0.3	0.2239
TS6.9	2		0.0909	0.0679	0.3636	0.2714	0.1818	0.1357	0.1818	0.1357	0.1818	0.1357
TS6.10	1		0	0.0000	0.2	0.0746	0.4	0.1493	0.2	0.0746	0.2	0.0746
TS6.11	1		0.2	0.0746	0.2	0.0746	0.2	0.0746	0.2	0.0746	0.2	0.0746
TS6.12	3		0.2	0.2239	0.2	0.2239	0.2	0.2239	0.2	0.2239	0.2	0.2239
TS6.13	3		0.2143	0.2399	0.2143	0.2399	0.2143	0.2399	0.2143	0.2399	0.1429	0.1599
TS6.14	1		0.25	0.0933	0	0	0.25	0.0933	0.25	0.0933	0.25	0.0933
TS6.15	2		0.2	0.1493	0.2	0.1493	0.2	0.14928	0.2	0.1493	0.2	0.1493
TS6.16	2		0.1429	0.1066	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599
TS6.17	2		0	0.0000	0	0	0.5	0.3732	0.5	0.3732	0	0
TS6.18	2		0	0.0000	0	0	0.2857	0.2133	0.4286	0.3199	0.2857	0.2133
TS6.19	2		0.1111	0.0829	0	0	0.3333	0.2488	0.3333	0.2488	0.2222	0.1659
TS6.20	8		0.0833	0.2488	0.25	0.7464	0.25	0.7464	0.25	0.7464	0.1667	0.4976
TS6.21	2		0.0833	0.0622	0.1667	0.1244	0.25	0.1866	0.25	0.1866	0.25	0.1866
TS6.22	2		0.2727	0.2036	0	0	0.2727	0.2036	0.1818	0.1357	0.2727	0.2036
TS6.23	2		0.3	0.2239	0	0	0.3	0.22392	0.3	0.2239	0.1	0.0746
TS6.24	2		0.2143	0.1599	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599	0.1429	0.1066
TS6.25	1		0.2727	0.1018	0	0	0.2727	0.1018	0.1818	0.0679	0.2727	0.1018
TS6.26	1		0.2727	0.1018	0	0	0.2727	0.1018	0.1818	0.0679	0.2727	0.1018
TS6.27	5		0	0.0000	0.2727	0.5089	0.2727	0.5089	0.2727	0.5089	0.1818	0.3393
TS6.28	2		0.0769	0.0574	0.2308	0.1722	0.2308	0.1722	0.2308	0.1722	0.2308	0.1722

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS6.29	1		0.1111	0.0415	0.3333	0.1244	0.1111	0.0415	0.3333	0.1244	0.1111	0.0415
TS6.30	2		0.1429	0.1066	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599	0.2143	0.1599
TS6.31	2		0.0833	0.0622	0.25	0.1866	0.25	0.1866	0.1667	0.1244	0.25	0.1866
TS6.32	1		0.0769	0.0287	0.2308	0.0861	0.2308	0.0861	0.2308	0.0861	0.2308	0.0861
TS6.33	2		0.1429	0.1066	0.4286	0.3199	0.1429	0.1066	0.1429	0.1066	0.1429	0.1066
TS6.34	1		0.3333	0.1244	0	0	0.3333	0.1244	0	0	0.3333	0.1244
TS6.35	2		0.1	0.0746	0	0	0.3	0.22392	0.3	0.2239	0.3	0.2239
TS6.36	1		0.1111	0.0415	0	0	0.2222	0.0829	0.3333	0.1244	0.3333	0.1244
TS6.37	1		0	0.0000	0.25	0.0933	0.25	0.0933	0.25	0.0933	0.25	0.0933
TS6.38	1		0.0833	0.0311	0.25	0.0933	0.25	0.0933	0.1667	0.0622	0.25	0.0933
TS7		0.0596										
TS7.1	5		0.2143	0.0639	0.2143	0.0639	0.2143	0.0639	0.2143	0.0639	0.1429	0.0426
TS7.2	3		0.1667	0.0298	0	0	0.5	0.0894	0.1667	0.0298	0.1667	0.0298
TS7.3	8		0.2308	0.1100	0.2308	0.1100	0.2308	0.1100	0.2308	0.1100	0.0769	0.0367
TS7.4	2		0.2	0.0238	0.2	0.0238	0.2	0.0238	0.2	0.0238	0.2	0.0238
TS7.5	5		0.1111	0.0331	0.2222	0.0662	0.2222	0.0662	0.2222	0.0662	0.2222	0.0662
TS7.6	2		0.2	0.0238	0	0	0.4	0.0477	0.4	0.0477	0	0
TS7.7	2		0.1429	0.0170	0.2143	0.0255	0.2143	0.0255	0.2143	0.0255	0.2143	0.0255
TS7.8	2		0.2	0.0238	0.2	0.0238	0.2	0.0238	0.2	0.0238	0.2	0.0238
TS7.9	1		0.2308	0.0138	0.2308	0.0138	0.1538	0.0092	0.1538	0.0092	0.2308	0.0138
TS7.10	1	0.0596	0.2	0.0119	0.2	0.0119	0.2	0.0119	0.2	0.0119	0.2	0.0119
TS7.11	2		0.2143	0.0255	0.2143	0.0255	0.2143	0.0255	0.2143	0.0255	0.1429	0.0170
TS7.12	2		0.1	0.0119	0.3	0.0358	0.3	0.0358	0.1	0.0119	0.2	0.0238
TS7.13	1		0.0909	0.0054	0.1818	0.0108	0.2727	0.0163	0.1818	0.0108	0.2727	0.0163

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS7.14	1		0.2222	0.0132	0	0	0.3333	0.0199	0.1111	0.0066	0.3333	0.0199
TS7.15	1		0	0.0000	0.6	0.0358	0	0	0	0	0.4	0.0238
TS7.16	1		0.2143	0.0128	0.2143	0.0128	0.2143	0.0128	0.2143	0.0128	0.1429	0.0085
TS7.17	5		0.0435	0.0130	0.2609	0.0777	0.2609	0.0777	0.1739	0.0518	0.2609	0.0777
TS7.18	2		0.0588	0.0070	0.2353	0.0280	0.3529	0.0421	0.1176	0.0140	0.2353	0.0280
TS7.19	1		0.0526	0.0031	0	0	0.3158	0.0188	0.3158	0.0188	0.3158	0.0188
TS7.20	5		0.2143	0.0639	0.2143	0.0639	0.1429	0.0426	0.2143	0.0639	0.2143	0.0639
TS7.21	2		0.0769	0.0092	0.2308	0.0275	0.2308	0.0275	0.2308	0.0275	0.2308	0.0275
TS7.22	8		0.2	0.0954	0.2	0.0954	0.2	0.0954	0.2	0.0954	0.2	0.0954
TS7.23	2		0.0909	0.0108	0.2727	0.0325	0.2727	0.0325	0.2727	0.0325	0.0909	0.0108
TS7.24	1		0.1429	0.0085	0.4286	0.0255	0.2857	0.0170	0.1429	0.0085	0	0
TS7.25	2		0.0769	0.0092	0.2308	0.0275	0.2308	0.0275	0.2308	0.0275	0.2308	0.0275
TS8		0.0818										
TS8.1	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327
TS8.2	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327
TS8.3	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327
TS8.4	8		0.1429	0.0935	0.2143	0.1402	0.2143	0.1402	0.2143	0.1402	0.2143	0.1402
TS8.5	8		0.2	0.1309	0.2	0.1309	0.2	0.1309	0.2	0.1309	0.2	0.1309
TS8.6	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327
TS8.7	1		0.2	0.0164	0.2	0.0164	0.2	0.0164	0.2	0.0164	0.2	0.0164
TS8.8	5		0.2	0.0818	0.2	0.0818	0.2	0.0818	0.2	0.0818	0.2	0.0818
TS8.9	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327
TS8.10	8		0.2	0.1309	0.2	0.1309	0.2	0.1309	0.2	0.1309	0.2	0.1309
TS8.11	2		0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327	0.2	0.0327

Criteria	Priority	Weighted Factor	RM_I		RM_III		RM_IV		RM_V		RM_VII	
			Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score	Norma. Value	Weighted Score
TS8.12	1		0.25	0.0205	0.1667	0.0136	0.0833	0.0068	0.25	0.0205	0.25	0.0205
TS8.13	2		0.1111	0.0182	0.2222	0.0364	0.2222	0.0364	0.2222	0.0364	0.2222	0.0364
TS8.14	1		0	0.0000	0.25	0.0205	0.25	0.0205	0.25	0.0205	0.25	0.0205
TS8.15	2		0.2	0.0327	0.3	0.0491	0.3	0.0491	0.2	0.0327	0	0
TS8.16	2		0.1429	0.0234	0	0	0.4286	0.0701	0.4286	0.0701	0	0
TS8.17	2		0.2857	0.0467	0.4286	0.0701	0.2857	0.0467	0	0	0	0
TS8.18	2		0.25	0.0409	0.25	0.0409	0.2500	0.0409	0.25	0.0409	0	0
TS8.19	2		0.1818	0.0297	0.2727	0.0446	0.2727	0.0446	0.2727	0.0446	0	0
TS8.20	2		0.1429	0.0234	0.2857	0.0467	0.2857	0.0467	0.2857	0.0467	0	0
TS8.21	1		0	0.0000	0.375	0.0307	0.25	0.0205	0.375	0.0307	0	0
TS8.22	1		0.5	0.0409	0.3333	0.0273	0.1667	0.0136	0	0	0	0
TS8.23	1		0	0.0000	0.25	0.02045	0	0	0.75	0.06135	0	0
TS9		0.0155										
TS9.1	5		0.2	0.0155	0.2	0.0155	0.2	0.0155	0.2	0.0155	0.2	0.0155
TS9.2	2		0.2308	0.0072	0.2308	0.0072	0.2308	0.0072	0.0769	0.0024	0.2308	0.0072
TS9.3	1		0.5	0.0078	0	0	0	0	0.5	0.0078	0	0
TS9.4	2		0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062
TS9.5	2		0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062
TS9.6	1		0.2	0.0031	0.2	0.0031	0.2	0.0031	0.2	0.0031	0.2	0.0031
TS9.7	2		0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062	0.2	0.0062
TS9.8	1		0.25	0.0039	0.25	0.0039	0	0	0.25	0.0039	0.25	0.0039
TS9.9	1		0.25	0.0039	0.25	0.0039	0	0	0.25	0.0039	0.25	0.0039
TS9.10	1		0.2	0.0031	0.2	0.0031	0.2	0.0031	0.2	0.0031	0.2	0.0031
TS9.11	2		0.1818	0.0056	0.1818	0.0056	0.1818	0.0056	0.2727	0.0085	0.1818	0.0056

Appendix E. Sensitivity index calculations for sensitivity analysis

Table E.3. Sensitivity index calculations for sensitivity analysis

Criteria	Sensitivity Variable +20%					
	RM_IV			RM_V		
	Δx	Δy	S index	Δx	Δy	S index
TS1.2	0.2	0.0300	0.1498	0	0	0
TS1.3	0.2	0.0068	0.0341	0.4	0.0124	0.0309
TS1.4	0.2	0.0131	0.0656	0.4	0.0237	0.0592
TS1.6	0.2	0.0066	0.0328	0.4	0.0118	0.0296
TS1.8	0.2	0.0029	0.0144	0.6	0.0076	0.0126
TS1.9	0.6	0.0212	0.0353	0.4	0.0112	0.0280
TS1.21	0.6	0.0092	0.0153	0.4	0.0047	0.0116
TS1.25	0.6	0.0184	0.0306	0.4	0.0093	0.0233
TS1.27	0.6	0.0083	0.0138	0.4	0.0040	0.0101
TS1.28	0.2	0.0054	0.0271	0.4	0.0095	0.0239
TS1.29	0.4	0.0065	0.0163	0.6	0.0084	0.0140
TS1.30	0.3	0.0116	0.0388	0.2	0.0063	0.0315
TS1.31	0.3	0.0106	0.0353	0.2	0.0056	0.0280
TS2.1	0.3	0.0051	0.0171	0.6	0.0087	0.0146
TS2.2	0.3	0.0022	0.0073	0.6	0.0036	0.0060
TS3.4	0.4	0.0365	0.0913	0.5	0.0354	0.0709
TS3.8	0.2	0.0938	0.4688	0.6	0.2515	0.4192
TS3.9	0.6	0.2315	0.3859	0.4	0.1132	0.2830
TS3.10	0.2	0.0938	0.4688	0.6	0.2515	0.4192
TS3.13	0.2	0.0590	0.2948	0	0.0000	0.0000
TS3.15	0.2	0.0261	0.1303	0.6	0.0707	0.1179
TS3.16	0.6	0.0402	0.0670	0.4	0.0165	0.0413
TS3.18	0.4	0.1415	0.3537	0	0.0000	0.0000
TS3.20	0.6	0.0742	0.1237	0.4	0.0392	0.0979
TS3.22	0.3	0.2001	0.6669	0.2	0.0922	0.4611
TS3.23	0.2	0.0393	0.1965	0	0.0000	#DIV/0!
TS3.27	0.6	0.0243	0.0405	0.2	0.0039	0.0197
TS3.32	0.6	0.0276	0.0459	0.4	0.0081	0.0202
TS3.33	0.6	0.0106	0.0176	0.5	0.0036	0.0071
TS3.36	0.6	0.0289	0.0482	0.4	0.0141	0.0354

Criteria	Sensitivity Variable +20%					
	RM_IV			RM_V		
	Δx	Δy	S index	Δx	Δy	S index
TS4.9	0.2	0.0257	0.1287	0.4	0.0446	0.1116
TS4.10	0.6	0.0154	0.0256	0.4	0.0063	0.0158
TS4.11	0.3	0.0693	0.2310	0	0.0000	0.0000
TS4.12	0.2	0.0000	0.0000	0	0.0000	0.0000
TS4.13	0.6	0.0491	0.0819	0.4	0.0249	0.0623
TS4.15	0.3	0.0098	0.0328	0.6	0.0163	0.0271
TS4.21	0.6	0.0450	0.0750	0.5	0.0295	0.0590
TS5.1	0.2	0.0556	0.2782	0.6	0.1537	0.2562
TS5.14	0.3	0.0229	0.0763	0.6	0.0299	0.0498
TS5.15	0.3	0.0455	0.1516	0.6	0.0751	0.1252
TS6.3	0.5	0.1493	0.2986	0.6	0.1466	0.2443
TS6.4	0.6	0.1390	0.2317	0.2	0.0332	0.1659
TS6.6	0.3	0.1583	0.5278	0.2	0.0730	0.3649
TS6.7	0.4	0.1448	0.3620	0.6	0.1860	0.3100
TS6.8	0.4	0.1103	0.2758	0.6	0.1344	0.2239
TS6.10	0.4	0.0804	0.2010	0.2	0.0249	0.1244
TS6.18	0.4	0.0464	0.1159	0.6	0.0384	0.0640
TS6.22	0.6	0.2036	0.3393	0.4	0.1031	0.2578
TS6.25	0.6	0.1018	0.1696	0.4	0.0516	0.1289
TS6.26	0.6	0.1018	0.1696	0.4	0.0516	0.1289
TS6.29	0.2	0.0371	0.1855	0.6	0.0995	0.1659
TS6.31	0.6	0.2205	0.3675	0.4	0.1144	0.2861
TS6.34	0.2	0.2488	1.2440	0	0.0000	0.0000
TS6.36	0.4	0.0469	0.1172	0.6	0.0547	0.0912
TS6.38	0.6	0.1103	0.1838	0.4	0.0572	0.1431
TS7.2	0.6	0.0368	0.0614	0.2	0.0060	0.0298
TS7.12	0.6	0.0484	0.0806	0.2	0.0119	0.0596
TS7.13	0.6	0.0163	0.0271	0.4	0.0082	0.0206
TS7.14	0.6	0.0222	0.0370	0.2	0.0053	0.0265
TS7.17	0.6	0.0848	0.1413	0.4	0.0435	0.1088
TS7.18	0.6	0.0421	0.0701	0.2	0.0098	0.0491
TS7.20	0.4	0.0611	0.1527	0.6	0.0792	0.1320

Criteria	Sensitivity Variable +20%					
	RM_IV			RM_V		
	Δx	**Δy**	**S index**	**Δx**	**Δy**	**S index**
TS7.24	0.4	0.0196	0.0491	0.2	0.0074	0.0369
TS8.12	0.2	0.0104	0.0520	0.6	0.0286	0.0477
TS8.15	0.6	0.0402	0.0669	0.4	0.0196	0.0491
TS8.17	0.4	0.1169	0.2921	0	0.0000	0.0000
TS8.21	0.4	0.0080	0.0200	0.6	0.0086	0.0143
TS8.22	0.2	0.0682	0.3408	0	0.0000	0.0000
TS8.23	0	0.0000	0.0000	0.6	0.0041	0.0068
TS9.2	0.6	0.0147	0.0245	0.2	0.0038	0.0191
TS9.3	0	0.0000	0.0000	0.6	0.0047	0.0078
TS9.8	0	0.0000	0.0000	0.6	0.0085	0.0142
TS9.9	0	0.0000	0.0000	0.6	0.0085	0.0142
TS9.11	0.4	0.0051	0.0129	0.6	0.0064	0.0107

UDK 004.414.2+004.2](043.3)

SL 344. 2023-01-30, 22,5 leidyb. apsk. l. Tiražas 14 egz. Užsakymas 21.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas