

# DD-KARB: Data-Driven Compliance to Quality by Rule Based Benchmarking

Mohammad Reza Besharati (✉ [besharati@ce.sharif.edu](mailto:besharati@ce.sharif.edu))

Sharif University of Technology

Mohammad Izadi

Sharif University of Technology

---

## Research Article

**Keywords:** compliance checking of systems, software quality, semantic logic, benchmarking, messenger apps, big data

**Posted Date:** January 7th, 2022

**DOI:** <https://doi.org/10.21203/rs.3.rs-1203650/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# Abstract

Rather than proof and logical evaluation, benchmarking can be used for compliance assessment. Naturally, a set of benchmarks can shape an applied solution to compliance assessment. This paper proposes the KARB solution system, i.e. preventing compliance anomalies through rule-based benchmarking. In fact, rule-based benchmarking means evaluating an under-compliance system with its symbolic specification and using a set of symbolic rules (on the behalf of the semantic logic of evaluation). A case study was conducted to demonstrate and analyze the KARB solution. The IR-QUMA study (Iranian Survey on Quality in Messenger Apps) was then conducted to evaluate the quality of some messenger applications. According to the evaluation results, the hybrid DD-KARB method (with a combination of semantics-awareness and data-drivenness) is more effective than solo methods and can compute a somehow good estimation for the messenger application user quality scores. Therefore, DD-KARB can be considered a method for quality benchmarking in this technical context.

## Introduction

Sharing a theoretical nature, many compliance assessment approaches are based on a systematic, rigorous, and formal theory of proof or evaluation (such as logics, formal languages, proof systems, reference models, and domain models). Theories have high levels of internal integrity but usually suffer from a lack of support and adaptation to the diversity and complexity of real-world cases. Sometimes, an aggregation of multiple simpler tools is more successful than a single, rigid, unified, in-depth-designed, and sophisticated tool.

In some real-world compliance-solving cases, more lightweight approaches to formal specification that support semantic modeling (*e.g.* generative grammars, production rules, set-theoretic notations, and rewriting logics) can play central roles in overcoming semantic diversity and complexity. These approaches support a kind of semantic compilation of diverse, domain-specific semantic models. For instance, generative rules can be set to define the mapping and composition logic of different, independent semantic models.

## Compliance Checking

Compliance solutions concern assessment, evaluation, verification, validation, and checking of systems, services, processes, products, designs, organizations, or environments with regard to rules, regulations, laws, standards, specifications, policies, guidelines, protocols, methods, principals, and reference-models [1][2]. Many application domains need and use compliance solutions including organizations and corporates in the following domains: software and IT industry [3], e-governance [2], finance and banking [4], legal sectors and professions [5], commerce and trade [6], highly regulated industries (*e.g.* food [7] and drug, medical services and devices [8], and construction industry [9]), complex and interdisciplinary products and services [10], emerging technology products and services (*e.g.* cyber-physical systems [11], self-driving cars [12], cognitive robotics and agents [13], and smart applications [14]).

There are many prominent compliance concerns that have been considered by numerous regulations, standards, laws, and acts. The most important concerns are as follows: security [15], safety [16][17][18], privacy [19][20], data protection [21], accountability [22], responsibility [23], transparency [24], competency [25], anti-piracy [26], anti-corruption [27], antitrust [27], accessibility [28], HCI, quality management and assurance [29][30][31], environmental management [32], sustainability [33], usability [34], human comfort [35], ethics [36], conformance with the disabilities [37], adherence to the children [38], the elderly [39], simplicity [40], ease of use.

Modern paradigms have amplified the necessity of compliance requirements (paradigms such as standardization in business, automation in industries, artificial intelligence and ubiquitous computing in society, complex systems engineering, sociotechnical systems, ongoing growth in the economy, social complexity, and quality maturity of services/processes).

In [41], a formal definition (as a 4-tuple) was presented for a special kind of benchmarking. There are a few formally-defined frameworks for compliance checking in legal applications, which are defined in theoretical manners such as formal systems [5] as well as conceptual modeling of legal texts [5]. Grammar-like and production-rule formalisms have been suggested for automated compliance checking in legal applications [42] [43] [44] [45].

Rules and grammars for architecture conformance checking, especially for “software quality assurance” [46], is another application domain. Some rule-based approaches for architecture selection relate the nonfunctional requirements, domain requirements, and quality characteristics to architectural styles [47], architectural models [48], architectural patterns [49] and architectural aspects [50].

Circuits and flows are considered recurrent modeling approaches in systems engineering. Some researchers regard circuits and flows as a basis for compliance modeling, checking, and benchmarking [51] [52]. There are numerous verification tools and solutions for flow-based models. These tools serve as a means of compliance checking. For instance, agent-coordination protocols for crisis situations could be modeled and checked by these tools and solutions [53].

In software engineering, there are some model-based approaches to compliance assurance [54]. These approaches employ a modeling notation or framework (*e.g.* UML, KAOS[55], and GSN[56]). A “model-based assurance case” is an approach to safety compliance management. It encompasses a compliance meta-model covering “claims” or “requirements”, “evidence”, “arguments”, and “contexts” [57] [58]. In [54], the need for a general model of compliance and compliance activity is addressed as an open-ended problem.

In [9] and [59], very close approaches were introduced. The meta-model and system architecture of these approaches are comparable with the one proposed in this study. There are some other meta-models for compliance checking applications and frameworks (See [60], [61], [62], and [3]).

# Benchmarking

A benchmark is the common or standard infrastructure employed to analyze, evaluate, and compare the reality of solutions, tools, or systems by their executions (for a few definitions, see [63], [64], [65], [66], [67], [68], [69], [70] and [71]. For some instances in other fields, see [72], [73], [74], [30], [75], and [76]). Sometimes, a measure can simply be used as a benchmark [77]. “Procedures”, “measures”, and “computers” are considered the most common concepts in diverse definitions of benchmarking (Figure 1).

(For some early attempts in the history of benchmarking in IT and computing, see [78], [79], [80], [81], and [82]). There is also a growing trend in benchmarking for quality assurance, management, and process improvement [83] [84]. It has been a progressive journey so far [85] [86] [87].

From a managerial standpoint, benchmarking requires a significant investment in time and perhaps money [84]. Hence, it should be considered a long-term profitable activity and a sort of infrastructure development for a field. In cloud computing, prior investments in performance measuring tools lead to the already available tools for the new field (for a case, see [88]). It was a chance; however, the dedicated attempts began for defining and developing benchmarks for cloud computing from scratch [89]. Moreover, investment in benchmarking is also important [90], for this decision can benefit all stakeholders [85].

Successful notions of benchmarking (in every field) are characterized by a community that creates, promotes, and uses benchmarks. Benchmarking can also be viewed as an applied manifestation and adoption of community knowledge and expertise [91].

There are informal guidelines for ensuring the quality of software (in terms of quality attributes such as security, integrity, and maintainability). Formal specification and automatic checking of these guidelines can contribute to the higher quality assurance of software (see [92] as an example for the formalization and automation of security guidelines).

Benchmarks can assess the quality (rather than only functionality) [29] [69] [71]; therefore, they are suitable for formal or systematic qualitative analysis of systems. For instance, security and compliance benchmarks have been reported [66] [93]. Measuring the productivity of an organization is another case that has qualitative dimensions (such as the level of customer satisfaction, the quality of products, or the extent to which an organization has the right group of staff [85]) which can be measured through some systematic approaches [85].

## Soft Benchmarks

A knowledge representation can be created once and then used many times — ontologies are a practical case of this manner of reusability (see [94]). Thus, a community can construct a knowledge

representation and use it as a standard and reusable asset. If this asset helps the community share their expertise, analyze their systems and solutions, and evaluate the behavior and other characteristics of their systems, it can then be considered a soft benchmark. The “soft” part of the title indicates its knowledge-related nature.

Knowledge representations are not limited to ontologies [95]. Formal specifications such as logical formulations, description logics, semantic networks, and rule-based approaches are considered alternatives [96]. Logical models have a share in compliance checking approaches. For instance, logical modeling of regulations is a method for rule representation and checking automation [97]. Rules can also be used as a paradigm for knowledge representation [98].

A logical theory for a piece of knowledge has the three essential characteristics of a benchmark. 1) It can be considered the common infrastructure due to the reusable and defined nature of a formal specification. 2) The results of reasoning indicate an examination and evaluation of the studied system and provide a basis for comparison between alternative and competing systems. 3) Executing reasoning on a logical theory of a piece of knowledge is an execution of meanings and semantics behind that knowledge. Thinking and mental activities can result in a hypothetical situation. In KARB, the rule-based reasoning schema can be viewed as a mimic of these natural procedures (the simulation of human auditing by automated and intelligent compliance audit tools would result in a proper need for the compliance industry [99], [66], and [2]).

Object models can act semantic models [100], especially for compliance checking purposes [101] [102]. For instance, Fornax objects capture specific rule semantics for the compliance checking of building designs [101]. These object models include contexts, domains, and sometimes system specifications [100]. Regarding Fornax, the objects for hospital design semantics differ from those for airport designs [101]. Software patterns are another representation form or media of technical knowledge. Pattern-based solutions to compliance checking have been addressed by some studies [103]. Compliance patterns are a kind of knowledge-capturing tool for compliance assessment.

In KARB, knowledge is represented through the intuitionistic formal semantics known as the “semantic logic”. The semantic logic was created to capture the semantics and meanings of text [40]. It is used in KARB for knowledge representation. Any knowledge has its semantics and meaning [104] [105] [106] [96]. The knowledge itself is captured if its semantics and meanings are captured.

Knowledge also has a specific structure [107]. Therefore, a meaning structure (or a semantic construct) could be a proper candidate for the manifestation of knowledge. Based on and adopted from [40], semantics and meanings are considered a constructs, lattices[1], or systems of realities (= intuitions). Any well-defined formulation of knowledge represents and refers to a combination of entities, things, objects, events, affairs, facts, physics, concepts, cognitions, affections, or any other sorts of basic realities and intuitions. Therefore, knowledge can aggregately and abstractly be considered a combination and construction of basic realities and intuitions (with a glue of operators such as logic, structures,

modalities, and any necessary ones). This manner of semantic definition is a constructive and intuitionistic one.

As an underlying philosophy in KARB, the reality and its meanings are composed of statics-related and dynamics-related meanings. Symbolic constructs capture the statics-related part of knowledge meanings, whereas the generative rules capture the dynamics-related part of knowledge meanings.

## The Karb Solution

The results of a rule-based benchmarking in KARB differ from those of other compliance assessment approaches. Other approaches yield the results in the form of “yes or no”, “correct or incorrect”, etc.; however, a pool of quantities (*i.e.* derived and generated symbols) is considered the results in KARB. Therefore, the overall state of working memory at the end of each benchmarking process indicates the evaluations of the studied compliance case, whereas a rigorous and reasoned evaluation with diverse dimensions and values would be achieved.

Every aspect of compliance concerns can be addressed with a separate rule-based benchmark. Every benchmark draws and adds a new simple line on the overall picture of compliance assessment scenes. A set of multiple, different, and diverse benchmarks can make an applied and realistic compliance assessment of a complex system. Relying on rule-based benchmarks, this experimental and applied approach to compliance assessment provides a new space for new sorts of innovative, creative, and diverse methods for compliance assessment.

Figure 3 illustrates a brief meta-model of KARB. The assessment of every compliance requirement is reified by a compliance benchmark, which in turn consists of some concrete rule-based benchmarks. Therefore, every compliance requirement declares meanings and semantics for a compliance benchmark that assesses it.

A **compliance symbol (CSYM)** abstracts a CCON in a similar sense of atom symbols in LISP, objects in OO languages (*e.g.* Java), and JSON fragments in NoSQL DBs, all of which are units for compositional parts. Every **compliance concern (CC)** is in association with some **compliance requirements<sup>[2]</sup> (CR)** which capture the notion and attitude of that concern. For instance, safety is a compliance concern which can be defined in a zoo as the following notion: the zoo animals must not be able to harm or threaten the visitors (see Example 1).

Some **compliance rules (CRUL)** aggregately define the operational realization of a **compliance requirement**. Every CRU defines a more rigorous, concrete, and special obligation than a CR. In KARB, the rules are considered to be finer than requirements. The overall shape of a requirement consists of the limiting lines of its constituting parts (= rules). Every **compliance rule** has some **compliance concepts (CCONs)** in its definition. In the computational mechanisms of KARB, a **compliance symbol (CSYM)**

abstracts a CCON. By using a glue of (logical, structural, modality, and any necessary) operators, a formal definition of a CRUL can be constructed from the CSYMs of its CCONs (see Example 1).

**Example 1:**

The System under Compliance	a Zoo
CC1	Safety
CR1	The zoo animals must not be able to harm or threaten the visitors.
CRUL1	The cage fences must have proper specifications and conditions.
CCONs	Cage, fence, proper specifications, proper conditions
CSYMs	CE, FE, PS, PC
Formal Specifications of CRUL1	$X \text{ [IS-A] FE(CE) ==> O[PS(X)] [AND] O[PC(X)]}$

In KARB, the manner of formal specification of a CRUL is based on the intuitionistic logic called the “semantic logic”. Technically, it can be viewed as an axiomatic system on symbols with Brouwer–Heyting–Kolmogorov interpretations for semantics [108]. Symbols are considered to be on the behalf of basic intuitions (concepts, entities, objects, things, events, values, quantities, qualities, etc.), whereas the studied system is viewed as a complex construction of basic intuitions.

Formally, it would be sufficient to consider the semantic logic consisting of 1) a set of symbols (on the behalf of basic intuitions) and 2) a set of rules on them. Every rule describes a symbol generation action. When its left-side symbolic structure is ready in the working memory, the right-side symbolic structure is generated and pushed to the lattice of symbols in the working memory (See Figure 2).

## Case Study: Software Quality Evaluation

Since the 1990s, there have been various approaches to defining measurable quality such as quality function deployment, goal question metrics, and software quality metrics [109]. These methods seek to shape a general and common framework for quality measurement concerns. However, some quality factors are contextual and user-dependent [110]. For instance, some studies have measured the quality attributes of messenger apps and services from the user perspective (See [111], [112], [113], and [114]) or based on user behavior (See [115], [116], and [117]).

Quality definitions can be seen as a hierarchical formal system of interrelated concepts [118] or attributes [119] [120]. This view helps create an explicitly defined conceptual construct for qualities, *i.e.* a concept-quantized definition of qualities. Hence, a quantification of qualities (which is a well-known but poorly-achieved goal for rigorous software engineering [121]) helps measure and perceive the true level of qualities in each application.

After definition, it is the time for operationalization. Every theoretical concept has its real instances on the ground. User feedback, comments, experiences, requests, requirements, desires, cognitions, and intuitions can help this grounding operationalization. Therefore, a good quality-definition theory needs a good quality-grounding theory.

There is a semantic gap between definition theories and grounding theories. The former has a neat nature, whereas the latter has a scruffy one. How is it possible to bridge the neat nature and the scruffy nature [96]? A glue model can come forward to resolve this challenge. This model must contain the main conceptual elements of both sides and try to relate them in a gradient conceptual spectrum. Since this is exactly the manner of the KARB solution, it can be used as a method of designing a software quality evaluation technique. It is also a kind of evaluation for KARB, for it demonstrates its usefulness for a real concern or problem in the software engineering community.

**Example 2:** The semantic logic is provided to model the semantics of this scenario.

*Using SMS-Based Dynamic Passwords for E-Banking Transactions:* This logic contains certain rules and intuitions from four context theories, *i.e.* mobile apps, deontic predicate logic, security and system (see Figure 4). The stateless model-checking of this scenario semantics (by symbolic-value generations) yields a “false” value; hence, there is a contradiction in the scenario. Figure 5 presents the explainable results in the proof construction lattice. Although there are some unmentioned reasoning operations, they are omitted for the sake of simplicity in this preliminary example.

## Dd-karb

In order to boost the model pragmatics, a methodic extension of the principal model of KARB is considered. The **Data-Driven KARB (DD-KARB)** incorporates data-calculated weights (based on Big Data gathered from people) and values to parametrize the rules. An example of a parametrized rule is presented below:

**Example 3:**

**Rule:**

$\alpha * (A \Rightarrow B) \Rightarrow \beta * (P(A) \Rightarrow P(B))$

**Interpretation:**

If the alpha instances of  $(A \Rightarrow B)$  are generated in the semantic solution to the system (or if the weight of  $(A \Rightarrow B)$  is equal to alpha), then the beta instances of  $(P(A) \Rightarrow P(B))$  must be generated by applying this rule.

Based on expert scores, data examinations, data schemas, AI pre-trained models, and other sources of data-driven models, the semantics-based KARB models can be parametrized, annotated, and enriched

with data-driven aspects. The full-fledged methods (by combining the data and semantics aspects) could be better than the solo methods. Each system context or domain of application has its own semantics and data. DD-KARB can be employed to record and adapt both data and semantics aspects. This manner of description or declaration can help define a hybrid semantic core for compliance checking and solving. A hybrid semantics can help approach some hard-to-check compliance requirements through automatic compliance-checking solutions.

## Ir-quma Study

The case study is a popular evaluation method in software engineering research. Case studies are frequently used in papers to demonstrate the capabilities of new techniques and methods [122]. A case study was conducted in order to demonstrate and analyze the manner of KARB solution. The IR-QUMA study (Iranian Survey on Quality in Messenger Apps) was defined to evaluate the quality of some messenger applications. It consists of these stages:

1. **Selecting messenger applications:** The selected applications were Telegram, WhatsApp, Eita, Soroush, Bale, and some other popular mobile messengers in the Iranian cyberspace. They were selected for the IR-QUMA case study due to the access to a large community of their users.
2. **Collecting data:** An online questionnaire was designed to collect the opinions of users and trace the specifications of user experiences. The seven main questions concerned “absolute quality”, “relative quality”, “user satisfaction”, “error-freeness”, “perceived UI complexity”, “rationality of routines”, and “accordance and usability”. The answer to each question ranged between 1 and 5 to represent choices from “very weak” to “excellent”. Figure 6 shows the running-average series of user responses (for a portion of dataset).
3. **Using the KARB solution:**
  - a. **Elicitation of involving semantic theories**
  - b. **Specification of involving semantic theories.** The KARB-based specifications were developed for each of the involving semantic theories. Figure 7 presents a detailed map of the involving semantic theories. The emphasis was given to these theories:
    - i. KARB-based specification of messenger apps
    - ii. KARB-based specification of some quality terms
    - iii. KARB-based specification of user behavior
    - iv. KARB-based specification of some pieces of HCI knowledge
    - v. KARB-based specification of risks and threats
    - vi. KARB-based specification of software platform and mechanisms
    - vii. KARB-based specification of cognitive aspects
    - viii. KARB-based specification of social aspects

c. **Computation and Model Checking:** The KARB solution was employed to compute some of the compliance anomalies.

4. **Evaluating the results.** The results were compared in three aspects: expert judgments, IT reports, and user opinions.

The IR-QUMA study details will be published in a separate report. However, this paper we used the collected data and the semantic model to conduct some experiments on deferent quality benchmarks, especially the KARB solution.

## Ir-quma Data Collection

A questionnaire was designed to evaluate some quality-related measures, metrics, and features from the user experience perspective. The questionnaire was published in the popular channels of Iranian mobile social networks on 10 different messengers (*i.e.* Telegram, WhatsApp, Instagram, Eita, Soroush, Bale, Gap, iGap, Shaad, and Rubika). More than 40 communities of users on these 10 messengers (which are shaping more than 350 micro-communities based on visiting hours and spatial partitions) contributed to this research questionnaire. The collected data exceeded 7k completed online forms (from more than 7k distinct participants). In the research dataset [123], for the sake of data privacy and protection reasons, the names of these messengers were hashed randomly by assigning the ID codes from M1 to M10.

Different sets of statistical analysis, time series analysis, frequency analysis, cluster analysis, classification analysis, geometry locus of data points, and topological data analysis were based on user opinion data to obtain useful insights. As an analysis example, the data were sorted in a temporal order (which conserved the segregation of micro-communities). A running average method was then adopted (with a window-size = 20). Therefore, 7k data points were obtained from different segments of those 350 micro-communities. Every micro-community with its segment-average had its own footprint in the total space of data-points. Moreover, every messenger app had its own footprint in the total space of data-points.

Figure 8 depicts a correlation locus analysis for two of quality measures for seven different messenger applications in 7k data segments of 350 micro-communities. Every point is in accordance with the measure values obtained from one segment of a micro-community. The blue points refer to the mentioned messengers, whereas the red points indicate the entire data space (for all studied messengers). Every axis demonstrates a 5-level measure value (obtained by averaging user opinions in one data segment).

Figure 9 demonstrates the analysis for two other measures, *i.e.* correctness *vs.* quality. Correctness means the error freeness and bug freeness of the messenger application. The results indicate that there is a buffer between “correctness increase/decrease” and “overall quality increase/decrease”. This means that the other factors (rather than correctness) can play a key role in the overall quality of software.

Figure 10 illustrates the histogram of score instances of user quality judgments for 10 applications. The topological analysis of these 10 curves indicates six different curve clusters based on change trends during five levels.

## Evaluation Of The Method

The KARB solution is based on the semantic logic specified in previous research steps and its performance to correctly calculate (or mimic) the user opinions (evaluated in terms of error percentage of benchmark-computed quality scores and user opinions about quality scores). The following experimentation setting was considered: four different methods for quality benchmarking and five different experiments (for five different messengers).

Every user opinion record involves two sections: 1) the user opinion about quality score, which was called the absolute quality score, and 2) the quality context. The quality context includes the factors that can affect or relate to user opinion about quality scores. Age, gender, and other data were gathered from the users stating their experiences with the messengers (the scores also included bug-freeness and error-freeness, perceived UI complexity, rationality of routines, score of usability, *etc.*). The value options for all scores in the questionnaires were defined in a 5-point Likert scale [124] (the Likert scale has been used in various domains of software engineering such as [125]). Figure 11 shows the structure of a user opinion record.

**Definition 1.**  $N_i =$  number of user opinion records about  $App_i$

**Definition 2.**  $Error\_Percentage(App_i, Method_j) =$

$$\frac{\sum_{k=1}^{N_i} |Benchmark\_Computed\_Quality(App_i, Method_j, Context_k) - User\_Quality\_Opinion_k|}{N_i}$$

Table 1 reports the evaluation results (in the above-mentioned experimentation setting). Accordingly, the data-driven KARB method reduced the error percentage significantly. Figure 12 shows the error reduction curves (for five experiments). The average curve for these five curves indicate a pseudo-Sigmoid form. In other words, the hybrid DD-KARB method (with combination of semantics-awareness and data-drivenness) is more effective than solo methods and can compute a somehow good estimation for messenger application user quality scores. Therefore, DD-KARB can be considered a method for quality benchmarking in this technical context.

**Table-1.** Evaluation results (for Experimentation-Plan-ID-1)

Experimentation-Plan-ID= 1			Error Percentage					
			Exp. 1 Telegram	Exp. 2 Eita	Exp. 3 Whatsapp	Exp. 4 Soroush	Exp. 5 Bale	Average
Type of Benchmark	Quantitative, Based on Stats	<b>IR-QUMA (Simple Average)</b>	18.4	24.3	26.3	31.8	30.1	26.2
	Qualitative, Based on Sense	<b>Expert Quality Scoring</b>	17.5	23.1	25.4	30.2	29.8	25.2
	Declarative, Based on Analysis	<b>KARB + Hill Climbing</b>	11.8	15.6	15.8	17.9	16.4	15.5
	Hybrid, Based on Solving	<b>Data-Driven KARB</b>	8.8	13.2	13.5	16.8	14.0	13.3

## Discussion And Conclusion

The first benchmark uses the simple average of the IR-QUMA context data as an estimator for quality scores, whereas the second benchmark is based on expert judgments about the quality of messengers. Moreover, the third benchmark is based on an initially voided-filled DD-KARB rules that obtain their weight values from a hill-climbing optimization algorithm for reaching a local minima of error. The last benchmark, but the first in precision, is based on the DD-KARB rules that obtain their weight values from two sources: 1) IR-QUMA data values and 2) a lightweight state-space-checking procedure for finding somehow good fitting parameters for the IR-QUMA dataset and the DD-KARB ruleset. Figure 13 depicts some results of this fitting procedure.

Therefore, the last benchmark incorporates these method features in a hybrid manner including semantics-awareness (by KARB), data-drivenness (by the DD part of DD-KARB and IR-QUMA data), and the fitting solution (by finding fitting parameters with a lightweight state-space-checking procedure).

The intended semantic landscape of this problem (*i.e.* quality measurement of messenger apps) involves more than 10 semantic theories. Without the semantic framework which the KARB solution provides, it would be impossible to focus on the most relevant parts of the wide semantic landscape of this complex problem. Without using the KARB rules which act as a kind of declarative dimensions in this problem, data-driven solving procedures and verification and model-checking methods would be unable to escape from the “state space explosion” [126] in this landscape.

However, the procedures and methods escaped with the help of KARB in DD-KARB. A 3-minute process on a conventional PC (Windows + Java + Intel Core i7 Processor) was successfully able to solve a somehow good fitting 10K-order complexity space (IR-QUMA) to a 10G-order value-space of weight values in the DD-KARB rules of the experiment.

It is concluded that the hybrid nature of the DD-KARB method (in the KARB Solution) can help solve some complex compliance problems in a lightweight manner and yield somehow good results (in terms of a low-error compliance-level quality estimator).

## Declarations

- Ethics approval and consent to participate: consents to participate were obtained.
- Consent for publication: not applicable.
- Availability of data and materials: M.R. Besharati, M. Izadi, IR-QUMA, 2020. <https://doi.org/http://dx.doi.org/10.17632/d89gphmnsk.3>.
- Competing interests: there is no competing interests.
- Funding: not applicable.
- Authors' contributions: Besharati: conceptualization, research, survey, meta-modeling, writing, reviewing and editing, data gathering, programming, visualization. Prof. Izadi: conceptualization, writing, reviewing and editing, supervision.
- Acknowledgements: Thanks to more than 7000 IR-QUMA survey participants.

## References

- [1] M.R. Besharati, M. Izadi, Semantics Based Compliance Solving, in: Fundam. Softw. Eng. (FSEN 2019), Student Poster Compet., 2019.
- [2] M. Hashmi, G. Governatori, H.P. Lam, M.T. Wynn, Are we done with business process compliance: state of the art and challenges ahead, *Knowl. Inf. Syst.* 57 (2018) 79–133.
- [3] O. Turetken, A. Elgammal, W.J. Van Den Heuvel, M.P. Papazoglou, Capturing compliance requirements: A pattern-based approach, *IEEE Softw.* 29 (2012) 28–36. <https://doi.org/10.1109/MS.2012.45>.
- [4] C. Brandt, F. Santini, N. Kokash, F. Arbab, Modeling and simulation of selected operational IT risks in the banking sector, in: ESM 2012 - 2012 Eur. Simul. Model. Conf. Model. Simul. 2012, 2012: pp. 192–200.
- [5] S. Ingolfo, A. Siena, A. Susi, A. Perini, J. Mylopoulos, Modeling laws with nomos 2, in: 2013 6th Int. Work. Requir. Eng. Law, RELAW 2013 - Proc., IEEE, 2013: pp. 69–71. <https://doi.org/10.1109/RELAW.2013.6671350>.
- [6] G. Governatori, Z. Milosevic, S. Sadiq, Compliance checking between business processes and business contracts, in: Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC, 2006: pp. 221–232. <https://doi.org/10.1109/EDOC.2006.22>.

- [7] A. Zasada, M. Fellmann, A Pattern-Based Approach to Transform Natural Text From Laws Into Compliance Controls in the Food Industry, *LWA*. 15 (2015).
- [8] S. Almpani, P. Stefaneas, H. Boley, T. Mitsikas, P. Frangos, A rule-based model for compliance of medical devices applied to the European market, *Int. J. Extrem. Autom. Connect. Healthc.* 1 (2019) 56–78.
- [9] J. Zhang, N.M. El-Gohary, Semantic-Based Logic Representation and Reasoning for Automated Regulatory Compliance Checking, *J. Comput. Civ. Eng.* 31 (2017) 04016037.  
[https://doi.org/10.1061/\(asce\)cp.1943-5487.0000583](https://doi.org/10.1061/(asce)cp.1943-5487.0000583).
- [10] S. Bragaglia, Monitoring complex processes to verify system conformance: A declarative rule-based framework, (2013).
- [11] S. Vuotto, Requirements-driven design of cyber-physical systems, in: *CEUR Workshop Proc.*, 2018.
- [12] C. Pek, V. Rusinov, S. Manzinger, M.C. Üste, M. Althoff., CommonRoad Drivability Checker: Simplifying the Development and Validation of Motion Planning Algorithms, in: *IEEE Intell. Veh. Symp.*, 2020: pp. 1013–1020.
- [13] B.O. Akinkunmi, F.M. Babalola, A norm enforcement mechanism for a time-constrained conditional normative framework, *Auton. Agent. Multi. Agent. Syst.* 34 (2020) 1–54. <https://doi.org/10.1007/s10458-020-09441-2>.
- [14] F.J. Miandashti, M. Izadi, A.A.N. Shirehjini, S. Shirmohammadi, An Empirical Approach to Modeling User-System Interaction Conflicts in Smart Homes, *IEEE Trans. Human-Machine Syst.* 50 (2020) 573–583.
- [15] S. Ranise, H. Siswantoro, Automated legal compliance checking by security policy analysis, in: *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2017: pp. 361–372. [https://doi.org/10.1007/978-3-319-66284-8\\_30](https://doi.org/10.1007/978-3-319-66284-8_30).
- [16] Elise GC Crawford, R.L. Kift, Keeping track of railway safety and the mechanisms for risk, *Saf. Sci.* 110 (2018) 195–205.
- [17] J.P. Castellanos Ardila, Facilitating Automated Compliance Checking of Processes against Safety Standards, Mälardalen University, 2019.
- [18] O. Kupferman, M.Y. Vardi., Model checking of safety properties, *Form. Methods Syst. Des.* 19 (2001) 291–314.
- [19] N.B. Truong, K. Sun, G.M. Lee, Y. Guo, GDPR-Compliant Personal Data Management: A Blockchain-Based Solution, *IEEE Trans. Inf. Forensics Secur.* 15 (2020) 1746–1761.  
<https://doi.org/10.1109/TIFS.2019.2948287>.

- [20] OCR, SUMMARY OF THE HIPAA PRIVACY RULE: HIPAA Compliance Assistance, Off. Civ. Rights. (2003).
- [21] O. Lynskey, *The foundations of EU data protection law*, Oxford University Press, 2015.
- [22] D. Butin, M. Chicote, D. Le Métayer, Log design for accountability, in: Proc. - IEEE CS Secur. Priv. Work. SPW 2013, IEEE, 2013: pp. 1–7. <https://doi.org/10.1109/SPW.2013.26>.
- [23] F.A. Bukhsh, P.G.G. Queiroz, Conceptual Modeling for Corporate Social Responsibility: A Systematic Literature Review, in: Econ. Grids, Clouds, Syst. Serv. 16th Int. Conf., 2019: p. 218.
- [24] R. Samavi, M.P. Consens, Publishing privacy logs to facilitate transparency and accountability, *J. Web Semant.* 50 (2018) 1–20.
- [25] S. Gay, T. Badrick, J. Ross., “State of the art” for competency assessment in Australian medical laboratories, *Accredit. Qual. Assur.* 25 (2020) 323–327.
- [26] J.P. Kesan, R.S. Gruner, *Intellectual Property Compliance: Systematic Methods for Building and Using Intellectual Property*, in: Cambridge Handb. Compliance, Cambridge University Press, 2020. <https://doi.org/10.2139/ssrn.3506951>.
- [27] R.M. Abrantes-Metz, E. Prewitt, Antitrust Compliance 2.0: The Use of Structural Analysis and Empirical Screens to Detect Collusion and Corruption in Bidding Procurement Processes, *Antitrust Chronicle, Compet. Policy Int.* (2015).
- [28] S.M. Baule, Evaluating the accessibility of special education cooperative websites for individuals with disabilities, *TechTrends.* 64 (2020) 50–56.
- [29] J.P. Correia, J. Visser, Benchmarking technical quality of software products, in: Proc. - 15th Work. Conf. Reverse Eng. WCRE’08, IEEE, 2008: pp. 297–300. <https://doi.org/10.1109/WCRE.2008.16>.
- [30] R. Baggen, J.P. Correia, K. Schill, J. Visser, Standardized code quality benchmarking for improving software maintainability, *Softw. Qual. J.* 20 (2012) 287–307. <https://doi.org/10.1007/s11219-011-9144-9>.
- [31] V. Lenarduzzi, F. Lomio, S. Moreschini, D. Taibi, D.A. Tamburri, Software Quality for AI: Where We Are Now?, in: *Int. Conf. Softw. Qual.*, 2021: pp. 43–53.
- [32] S. De Craemer, J. Vercauteren, F. Fierens, W. Lefebvre, F.J.R. Meysman, Using Large-Scale NO<sub>2</sub>Data from Citizen Science for Air-Quality Compliance and Policy Support, *Environ. Sci. Technol.* 54 (2020) 11070–11078. <https://doi.org/10.1021/acs.est.0c02436>.
- [33] C. Schreiber, Automated Sustainability Compliance Checking Using Process Mining and Formal Logic, in: *7th Int. Conf. ICT Sustain.*, 2020: pp. 181–184.

- [34] L. Alonso-Virgós, J.P. Espada, O.S. Martínez, R.G. Crespo, Compliance and application tests of usability guidelines about giving information quickly and comprehensibly, *Complex Intell. Syst.* (2020) 1–21.
- [35] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, R. Ke, Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving, *Transp. Res. Part C Emerg. Technol.* 117 (2020) 102662.
- [36] A. Christina, T.L. Fort, Finding the fit: Why compliance and ethics programs should seek to match individual and corporate values, *Bus. Horiz.* 63 (2020) 451–462.
- [37] A. Gavine, F. Spillers, Toward a Disability-Centric Model of User Participation in Accessibility Efforts: Lessons from a Case Study of School Children, in: *Int. Conf. Human-Computer Interact.*, 2020: pp. 76–86.
- [38] R. Tahir, F. Arif, A measurement model based on usability metrics for mobile learning user interface for children, *Int. J. E-Learning Educ. Technol. Digit. Media.* 1 (2015) 16–31.
- [39] L. Bajenaru, I.A. Marinescu, C. Dobre, G.I. Prada, C.S. Constantinou, Towards the development of a personalized healthcare solution for elderly: From user needs to system specifications, in: *Proc. 12th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2020*, IEEE, 2020: pp. 1–6.  
<https://doi.org/10.1109/ECAI50035.2020.9223254>.
- [40] M. Besharati, M. Izadi, Deciding About Semantic Complexity of Text By DAST Model, *ArXiv.* (2019).
- [41] A.N. Jacobvitz, A.D. Hilton, D.J. Sorin, Multi-program benchmark definition, in: *ISPASS 2015 - IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2015: pp. 72–82. <https://doi.org/10.1109/ISPASS.2015.7095786>.
- [42] T. V. V. V. Prasad, R.B. Korrapati, Computerized Applications of Natural Language Processing in Digital Economy: A Review, *Int. J. Eng. Manag. Res.* 7 (2017) 239–241.
- [43] E. Nash, J. Wiebensohn, R. Nikkilä, A. Vatsanidou, S. Fountas, R. Bill, Towards automated compliance checking based on a formal representation of agricultural production standards, *Comput. Electron. Agric.* 78 (2011) 28–37. <https://doi.org/10.1016/j.compag.2011.05.009>.
- [44] J.C. Maxwell, A.I. Antón, Developing production rule models to aid in acquiring requirements from legal texts, in: *Proc. IEEE Int. Conf. Requir. Eng.*, 2009: pp. 101–110. <https://doi.org/10.1109/RE.2009.21>.
- [45] S. Ingolfo, *Nomos 3: legal compliance of software requirements*, University of Trento, 2015.
- [46] L.R. De Silva, *Towards controlling software architecture erosion through runtime conformance monitoring*, University of St Andrews, 2014.
- [47] S. Moaven, H. Ahmadi, J. Habibi, A. Kamandi, Decision Support System Environment for Software Architecture Style Selection (DESAS v1. 0), in: *SEKE09*, 2009: pp. 147–151.

- [48] F. Nawaz, A. Mohsin, S. Fatima, N.K. Janjua, Rule-based multi-criteria framework for SaaS application architecture selection, in: IFIP Adv. Inf. Commun. Technol., 2015: pp. 129–138.  
[https://doi.org/10.1007/978-3-319-25261-2\\_12](https://doi.org/10.1007/978-3-319-25261-2_12).
- [49] A. Alebrahim, S. Fassbender, M. Filipczyk, M. Goedicke, M. Heisel, Towards systematic selection of architectural patterns with respect to quality requirements, in: ACM Int. Conf. Proceeding Ser., ACM, 2015: pp. 1–20. <https://doi.org/10.1145/2855321.2855362>.
- [50] M. del Mar Roldán-García, J. García-Nieto, A. Maté, J. Trujillo, J.F. Aldana-Montes, Ontology-driven approach for KPI meta-modelling, selection and reasoning, Int. J. Inf. Manage. (2019) 102018.  
<https://doi.org/10.1016/j.ijinfomgt.2019.10.003>.
- [51] C. Preidel, A. Borrmann, Refinement of the visual code checking language for an automated checking of building information models regarding applicable regulations, in: Congr. Comput. Civ. Eng. Proc., 2017: pp. 157–165. <https://doi.org/10.1061/9780784480823.020>.
- [52] N. Kokash, F. Arbab, Formal behavioral modeling and compliance analysis for Service-oriented systems, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2009: pp. 21–41. [https://doi.org/10.1007/978-3-642-04167-9\\_2](https://doi.org/10.1007/978-3-642-04167-9_2).
- [53] A. Nowroozi, P. Teymoori, T. Ramezanifarkhani, M.R. Besharati, M. Izadi, A Crisis Situations Decision-Making Systems Software Development Process with Rescue Experiences, IEEE Access. 8 (2020).  
<https://doi.org/10.1109/ACCESS.2020.2981789>.
- [54] S. Kokaly, R. Salay, M. Sabetzadeh, M. Chechik, T. Maibaum, Model management for regulatory compliance: A position paper, in: Proc. - 8th Int. Work. Model. Softw. Eng. MiSE 2016, ACM, 2016: pp. 74–80. <https://doi.org/10.1145/2896982.2896985>.
- [55] A. Van Lamsweerde, Requirements engineering: From system goals to UML models to software, John Wiley & Sons, 2009.
- [56] T. Kelly, R. Weaver, The Goal Structuring Notation-A Safety Argument Notation, in: Proc. Dependable Syst. Networks 2004 Work. Assur. Cases, 2004: p. 6.
- [57] R. Dardar, B. Gallina, A. Johnsen, K. Lundqvist, M. Nyberg, Industrial experiences of building a safety case in compliance with ISO 26262, in: Proc. - 23rd IEEE Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2012, 2012. <https://doi.org/10.1109/ISSREW.2012.86>.
- [58] T.P. Kelly, Arguing safety: a systematic approach to managing safety cases, University of York, 1999.
- [59] J. Zhang, N.M. El-Gohary, Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking, Autom. Constr. 73 (2017) 45–57.  
<https://doi.org/10.1016/j.autcon.2016.08.027>.

- [60] A. Siena, A. Perini, A. Susi, J. Mylopoulos, A meta-model for modelling law-compliant requirements, in: 2009 2nd Int. Work. Requir. Eng. Law, RELAW 2009, IEEE, 2009: pp. 45–51. <https://doi.org/10.1109/RELAW.2009.1>.
- [61] Christina Stratigaki, P. Loucopoulos, M. Nikolaidou, Designing a Meta Model as the Foundation for Compliance Capability, in: Int. Work. Capab. Bus. Informatics Organ. Conjunction with 16th IEEE Conf. Bus. Informatics, 2014.
- [62] M.P. Papazoglou, Making business processes compliant to standards & regulations, in: Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC, 2011: pp. 3–13. <https://doi.org/10.1109/EDOC.2011.37>.
- [63] W.J. Price, A Benchmark tutorial, IEEE Micro. 9 (1989) 28–43. <https://doi.org/10.1109/40.45825>.
- [64] D. Patterson, For better or worse, benchmarks shape a field: technical perspective, Commun. ACM. 55 (2012) 104–104.
- [65] R. Almeida, M. Vieira, Benchmarking the resilience of self-adaptive software systems: Perspectives and challenges, in: 6th Int. Symp. Softw. Eng. Adapt. Self-Managing Syst., 2011: pp. 190–195. <https://doi.org/10.1145/1988008.1988035>.
- [66] M. Whitman, H. Mattord, Readings & Cases in Information Security: Law & Ethics, Nelson Education, 2010.
- [67] K. Kanoun, L. Spainhower, Dependability benchmarking for computer systems, Wiley-IEEE Computer Society Press, 2008.
- [68] J. Friginal, D. De Andrés, J.C. Ruiz, P. Gil, Towards benchmarking routing protocols in wireless mesh networks, Ad Hoc Networks. 9 (2011) 1374–1388. <https://doi.org/10.1016/j.adhoc.2011.03.010>.
- [69] C. Le Goues, N. Holtschulte, E.K. Smith, Y. Brun, P. Devanbu, S. Forrest, W. Weimer, The ManyBugs and IntroClass Benchmarks for Automated Repair of C Programs, IEEE Trans. Softw. Eng. 41 (2015) 1236–1256. <https://doi.org/10.1109/TSE.2015.2454513>.
- [70] W.F. Tichy, Should computer scientists experiment more?, Computer (Long Beach, Calif). 31 (1998) 32–40. <https://doi.org/10.1109/2.675631>.
- [71] and R.M. Jesus Friginal, David de Andrés, Juan-Carlos Ruiz, Using dependability benchmarks to support ISO/IEC SQuaRE, in: 2011 IEEE 17th Pacific Rim Int. Symp. Dependable Comput., 2011.
- [72] C.A. Furia, M. Nordio, N. Polikarpova, J. Tschannen, AutoProof: auto-active functional verification of object-oriented programs, Int. J. Softw. Tools Technol. Transf. 19 (2017) 697–716. <https://doi.org/10.1007/s10009-016-0419-0>.
- [73] Y. Wang, Stream Processing Systems Benchmark: StreamBench, Aalto University, 2016.

- [74] B.W. Weide, M. Sitaraman, H.K. Harton, B. Adcock, P. Bucci, D. Bronish, W.D. Heym, J. Kirschenbaum, D. Frazier, Incremental benchmarks for software verification tools and techniques, in: *Verif. Softw. Theor. Tools, Exp. (Lecture Notes Comput. Sci., 2008: pp. 84–98. <https://doi.org/10.1007/978-3-540-87873-5-10>.*
- [75] S. Sattari, M. Izadi, An exact algorithm for the minimum dilation triangulation problem, *J. Glob. Optim.* 69 (2017) 343–367. <https://doi.org/10.1007/s10898-017-0517-x>.
- [76] J. Darmont, Data-Centric Benchmarking, *Encycl. Inf. Sci. Technol.* (2018) 1772–1782.
- [77] K. Arnett, G. Templeton, D. Vance, Information security by words alone: The case for strong security policies, *Int. J. Inf. Secur. Priv.* 3 (2009) 84–89.
- [78] E.O. JOSLIN, Describing workload for acquiring ADP equipment and software, *Comput. Autom.* 18 (1969) 36.
- [79] and J.J.A. Edward O. Joslin, The validity of basing computer selections on benchmark results, *Comput. Autom.* 15 (1966) 22–23.
- [80] J.R. Hillegass, Standardized benchmark problems measure computer performance, *Comput. Autom.* 15 (1966) 16–19.
- [81] E.O. Joslin, R.F. Chairman-Hitti., Evaluation and performance of computers: application benchmarks: the key to meaningful computer evaluations, in: *Proc. 1965 20th Natl. Conf., ACM, 1965: pp. 27–37.*
- [82] J. Yeh, A report on computer performance evaluation techniques, 1970.
- [83] R.C. Camp, Benchmarking: The search for industry best practices that lead to superior performance, in: *Benchmarking Search Ind. Best Pract. That Lead to Super. Perform., ASQC/Quality Resources, 1989. <https://doi.org/10.5860/choice.27-2173>.*
- [84] and S.J.K. Lawrence S. Pryor, How benchmarking goes wrong, *Plan. Rev.* 21 (1993) 6–53.
- [85] S. Mohapatra, Information theory and best practices in the IT industry, Springer Science & Business Media, 2012. <https://doi.org/10.1007/978-1-4614-3043-8>.
- [86] R. Scott, Benchmarking: A literature review, 2011.
- [87] Y.A. Adewunmi, R. Iyagba, M. Omirin, Multi-sector framework for benchmarking in facilities management, *Benchmarking.* 24 (2017) 826–856. <https://doi.org/10.1108/BIJ-10-2015-0093>.
- [88] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, *Commun. ACM.* 53 (2010) 50–58. <https://doi.org/10.1145/1721654.1721672>.

- [89] L. Gillam, B. Li, J. O'Loughlin, A.P.S. Tomar, Fair benchmarking for cloud computing systems, *J. Cloud Comput.* 2 (2013) 6. <https://doi.org/10.1186/2192-113X-2-6>.
- [90] M. Zeuch, *Handbook of human resources management*, 2016. <https://doi.org/10.1007/978-3-662-44152-7>.
- [91] S.E. Sim, S. Easterbrook, R.C. Holt, Using benchmarking to advance research: A challenge to software engineering, in: *Proc. - Int. Conf. Softw. Eng.*, 2003. <https://doi.org/10.1109/icse.2003.1201189>.
- [92] Z. Zhioua, Y. Roudier, R.A.-B.T. Kechiche, S. Short, Tracking Dependent Information Flows, in: *ICISSP*, 2017: pp. 179–189.
- [93] N. Antunes, M. Vieira, Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples, *IEEE Trans. Serv. Comput.* 8 (2015) 269–283. <https://doi.org/10.1109/TSC.2014.2310221>.
- [94] M. Landhäußer, S. Weigelt, W.F. Tichy, NLCl: a natural language command interpreter, *Autom. Softw. Eng.* 24 (2017) 839–861. <https://doi.org/10.1007/s10515-016-0202-1>.
- [95] N. Guarino, Formal ontology, conceptual analysis and knowledge representation, *Int. J. Hum. - Comput. Stud.* 43 (1995) 625–640. <https://doi.org/10.1006/ijhc.1995.1066>.
- [96] S. Russell, P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*, 2016. <https://doi.org/10.1017/S0269888900007724>.
- [97] C. Giblin, A.Y. Liu, S. Müller, B. Pfitzmann, X. Zhou, Regulations Expressed As Logical Models ( REALM ), *JURIX*. (2005) 37–48.
- [98] G.J. Nalepa, *Modeling with Rules Using Semantic Knowledge Engineering*, Springer, 2018.
- [99] F. Yip, A.K.Y. Wong, N. Parameswaran, P. Ray, Towards robust and adaptive semantic-based Compliance Auditing, in: *Proc. - IEEE Int. Enterp. Distrib. Object Comput. Work. EDOC, IEEE*, 2007: pp. 181–188. <https://doi.org/10.1109/EDOCW.2007.33>.
- [100] M. Fowler, *Domain-specific languages*, Pearson Education, 2010.
- [101] C. Eastman, J. min Lee, Y. suk Jeong, J. kook Lee, Automatic rule-based checking of building designs, *Autom. Constr.* 18 (2009) 1011–1033. <https://doi.org/10.1016/j.autcon.2009.07.002>.
- [102] A.S. Ismail, K.N. Ali, N.A. Iahad, A Review on BIM-based automated code compliance checking system, in: *Int. Conf. Res. Innov. Inf. Syst. ICRIS, IEEE*, 2017: pp. 1–6. <https://doi.org/10.1109/ICRIS.2017.8002486>.
- [103] A. Elgammal, O. Turetken, W.J. Van Den Heuvel, Using patterns for the analysis and resolution of compliance violations, in: *Int. J. Coop. Inf. Syst.*, 2012: pp. 31–54.

<https://doi.org/10.1142/S0218843012400023>.

[104] G.D. Bhatt, Knowledge management in organizations: Examining the interaction between technologies, techniques, and people, *J. Knowl. Manag.* 5 (2001) 68–75.

<https://doi.org/10.1108/13673270110384419>.

[105] D.H. Jonassen, R.M. Marra, Concept mapping and other formalisms as mindtools for representing knowledge, *ALT-J.* 2 (1994) 50–56. <https://doi.org/10.1080/0968776940020107>.

[106] H.C.G. Johnsen, *The new natural resource: Knowledge development, society and economics*, Routledge, 2016.

[107] A. Ortony, The representation of knowledge in memory, in: W.E.M. R.C. Anderson, R. J. Spiro (Ed.), *Sch. Acquis. Knowl.*, Lawrence Erlbaum Associates, 1977: pp. 99–135.

[108] M. Sato, Classical Brouwer-Heyting-Kolmogorov interpretation, in: *Algorithmic Learn. Theory*, Springer Berlin Heidelberg, 1997: pp. 176–196.

[109] V.R. Basili, G. Caldiera, H.D. Rombach, The goal question metric approach, *Encycl. Softw. Eng.* (1994) 528–532.

[110] H. Hu, S. Wang, C.P. Bezemer, A.E. Hassan, Studying the consistency of star ratings and reviews of popular free hybrid Android and iOS apps, *Empir. Softw. Eng.* (2019). <https://doi.org/10.1007/s10664-018-9617-6>.

[111] R. Arianto, F.L. Gaol, E. Abdurachman, Y. Heryadi, H.L.H.S. Warnars, B. Soewito, H. Perez-Sanchez, Quality measurement of android messaging application based on user experience in microblog, in: *Proc. - 2017 Int. Conf. Appl. Comput. Commun. Technol. ComCom 2017, IEEE, 2017*: pp. 1–5.

<https://doi.org/10.1109/COMCOM.2017.8167099>.

[112] C.H. Lien, Y. Cao, X. Zhou, Service quality, satisfaction, stickiness, and usage intentions: An exploratory evaluation in the context of WeChat services, *Comput. Human Behav.* 68 (2017) 403–410.

<https://doi.org/10.1016/j.chb.2016.11.061>.

[113] H. Hu, *Studying the Perceived Quality Consistency of Cross-Platform Mobile Apps*, Queen's University, 2017.

[114] C.M. Jongerius, *Quantifying Chatbot Performance by using Data Analytics*, Utrecht University, 2018.

[115] E. Vaziripour, R. Farahbakhsh, M. O'Neill, J. Wu, K. Seamons, D. Zappala, A Survey Of the Privacy Preferences and Practices of Iranian Users of Telegram, in: *Work. Usable Secur.*, 2018.

<https://doi.org/10.14722/usec.2018.23033>.

- [116] A. Hashemi, M.A. Zare Chahooki, Telegram group quality measurement by user behavior analysis, *Soc. Netw. Anal. Min.* 9 (2019) 33. <https://doi.org/10.1007/s13278-019-0575-9>.
- [117] H. Kermani, A. Mozaffari, The study of Iranian users' reasons in preferring Telegram on other Instant Messaging Applications, *MEDIA Stud.* 13 (2018) 7–20.
- [118] G. Büyüközkan, C.A. Havle, O. Feyzioğlu, A new digital service quality model and its strategic analysis in aviation industry using interval-valued intuitionistic fuzzy AHP, *J. Air Transp. Manag.* 86 (2020) 101817. <https://doi.org/10.1016/j.jairtraman.2020.101817>.
- [119] C. Yue, An intuitionistic fuzzy projection-based approach and application to software quality evaluation, *Soft Comput.* 24 (2020) 429–443.
- [120] M. Khan, M.D. Ansari, Multi-criteria software quality model selection based on divergence measure and score function, *J. Intell. Fuzzy Syst.* (2020) 1–10. <https://doi.org/10.3233/JIFS-191153>.
- [121] Y. Schneider, A. Busch, A. Koziolk, Using informal knowledge for improving software quality trade-off decisions, in: *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Springer, 2018: pp. 265–283. [https://doi.org/10.1007/978-3-030-00761-4\\_18](https://doi.org/10.1007/978-3-030-00761-4_18).
- [122] D.E. Perry, S.E. Sim, S.M. Easterbrook, Case studies for software engineers, in: *Proc. - 28th Int. Conf. Softw. Eng.*, 2006: pp. 1045–1046. <https://doi.org/10.1109/icse.2004.1317512>.
- [123] M.R. Besharati, M. Izadi, IR-QUMA, 2020. <https://doi.org/http://dx.doi.org/10.17632/d89gphmnsk.3>.
- [124] R. Likert, A technique for the measurement of attitudes, *Arch. Psychol.* (1932).
- [125] L. Gren, R. Torkar, R. Feldt, The prospects of a quantitative measurement of agility: A validation study on an agile maturity model, *J. Syst. Softw.* 107 (2015) 38–49. <https://doi.org/10.1016/j.jss.2015.05.008>.
- [126] M. Izadi, Model checking of component connectors, Leiden University, 2011.

## Figures

### Figure 1

A segmented word cloud showing the emphasis on different concepts in definitions of benchmarking

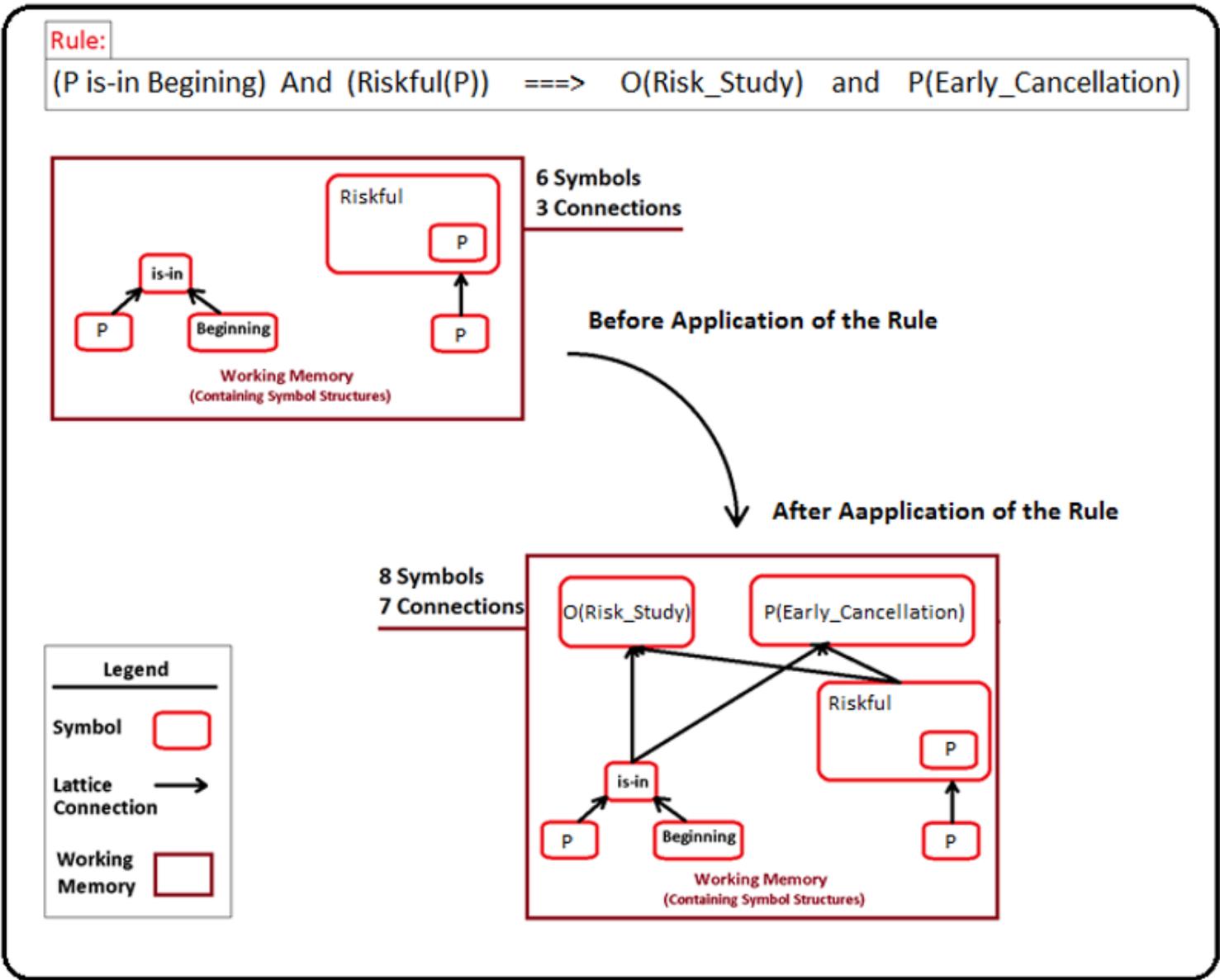
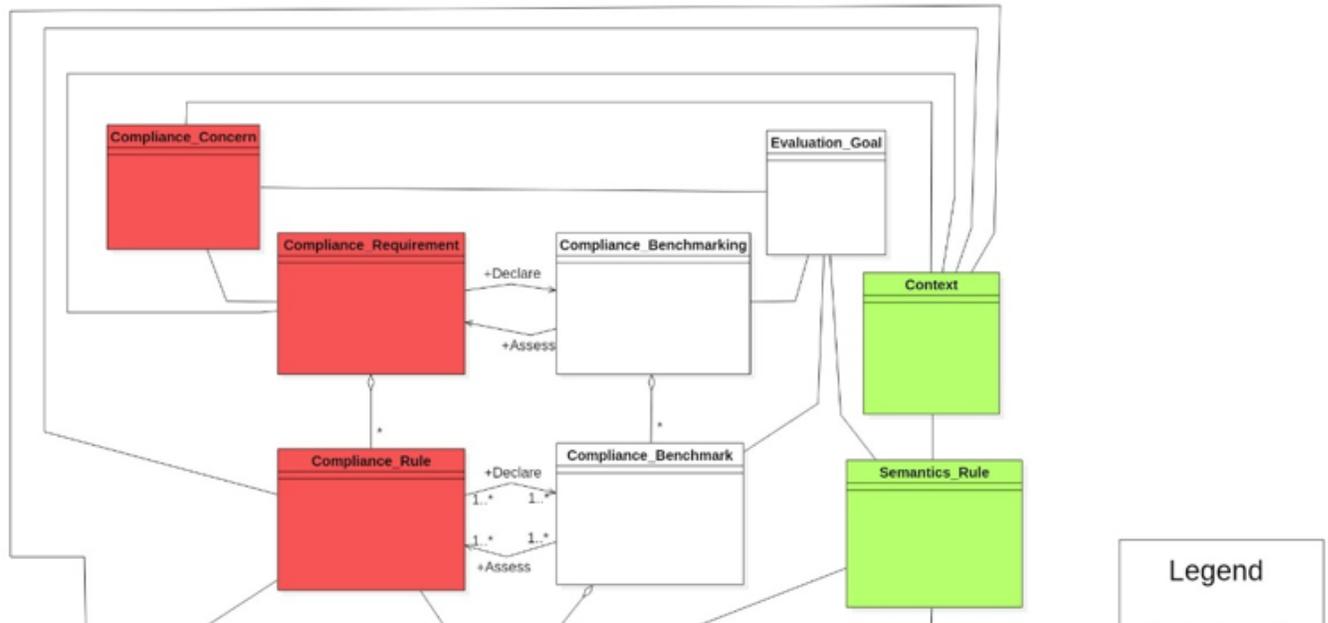


Figure 2

An instance of a semantic logic rule and effects of its application on working memory. The rule specifies the existence of Risk Study obligation and Early Cancellation probability when a project occurs in a riskful beginning [40].



**Figure 3**

The meta-model of the KARB solution is depicted as a class diagram[1] of elements. There are three main dimensions for KARB elements: compliance-related (red), benchmarking-related (white), and semantics-related (green) concepts.

[1] UML Class Diagram is a popular tool for concept modeling and meta-modeling.

**Figure 4**

The involving semantic theories and the semantic logic for Example 2

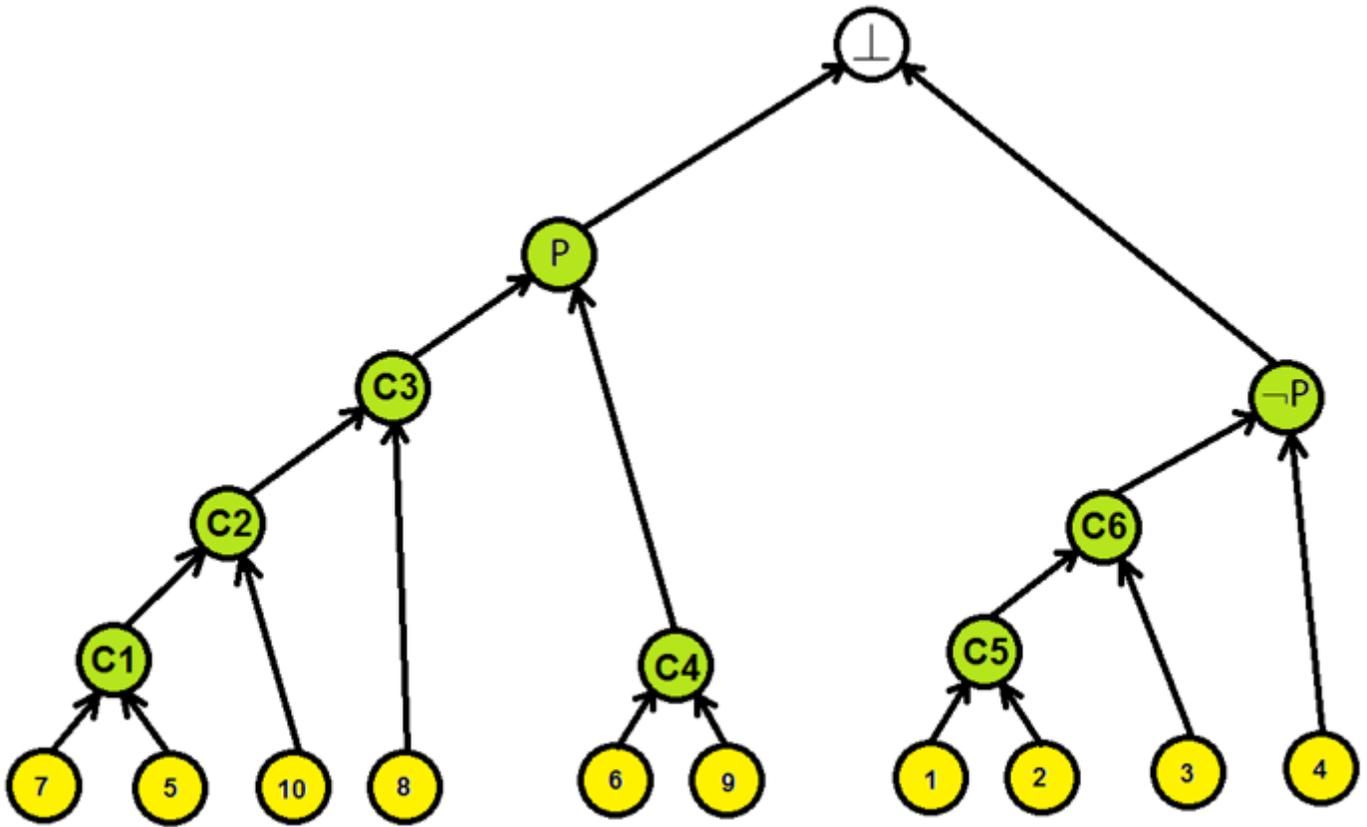
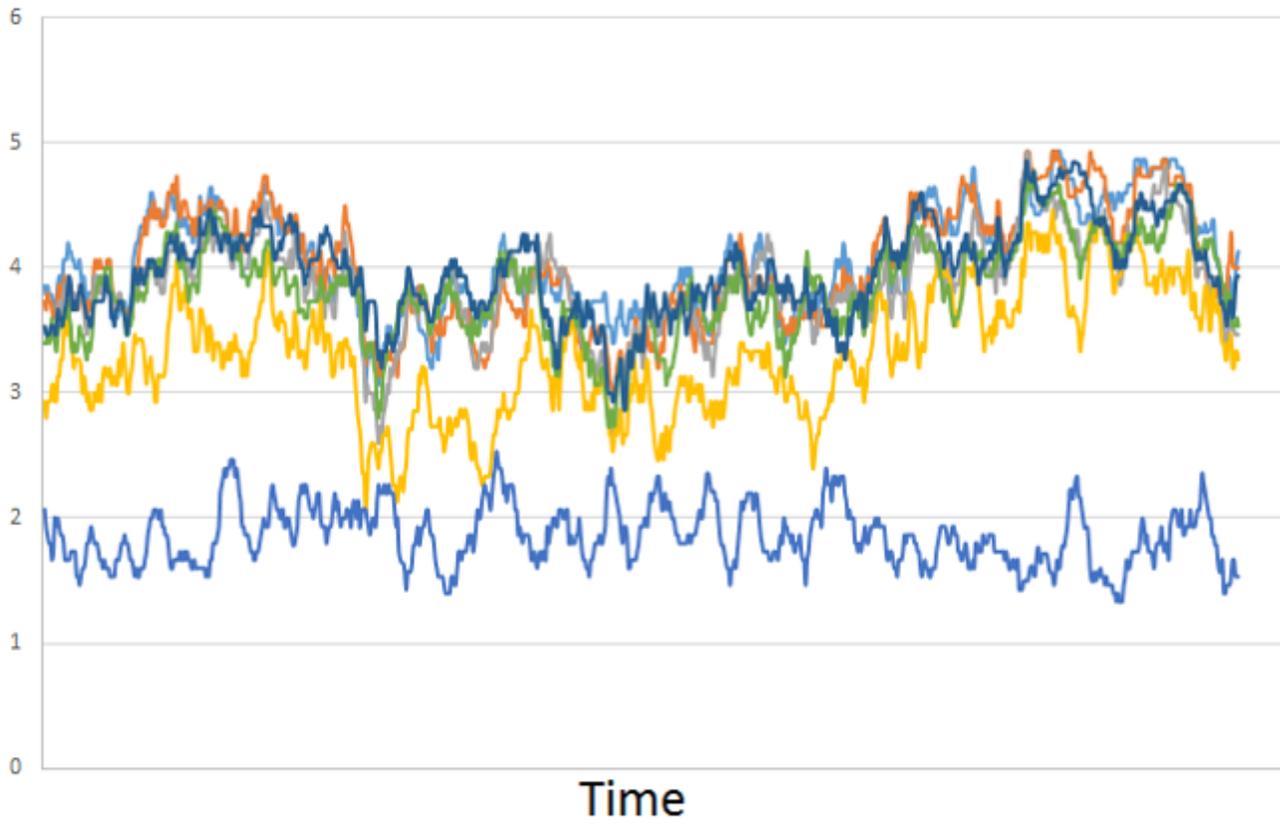


Figure 5

The proof construct lattice for false value from the scenario semantics

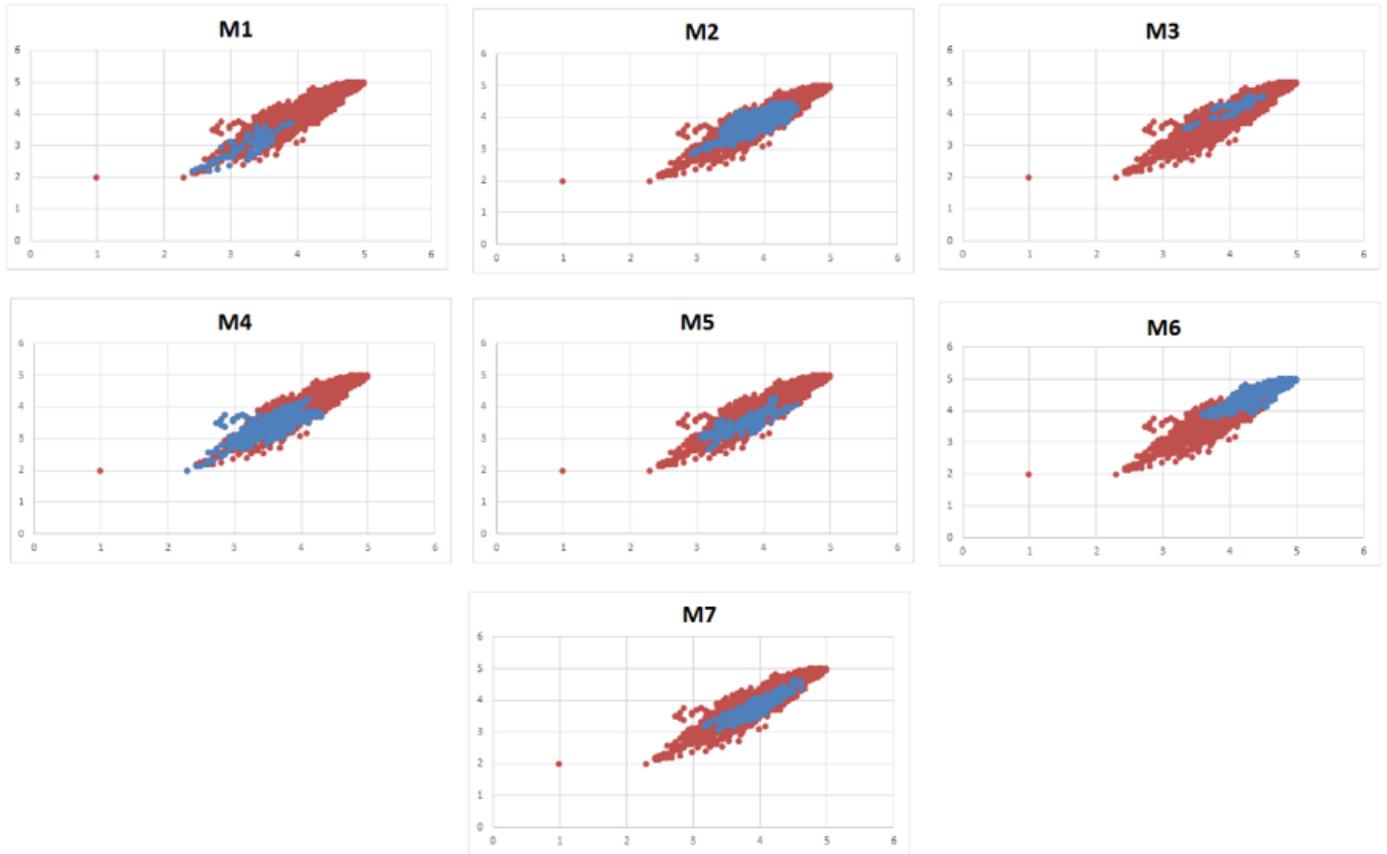


**Figure 6**

The running-average series of responses to 7 different questions in the questionnaire

**Figure 7**

A detailed map of involving semantic theories in the IR-QUMA study

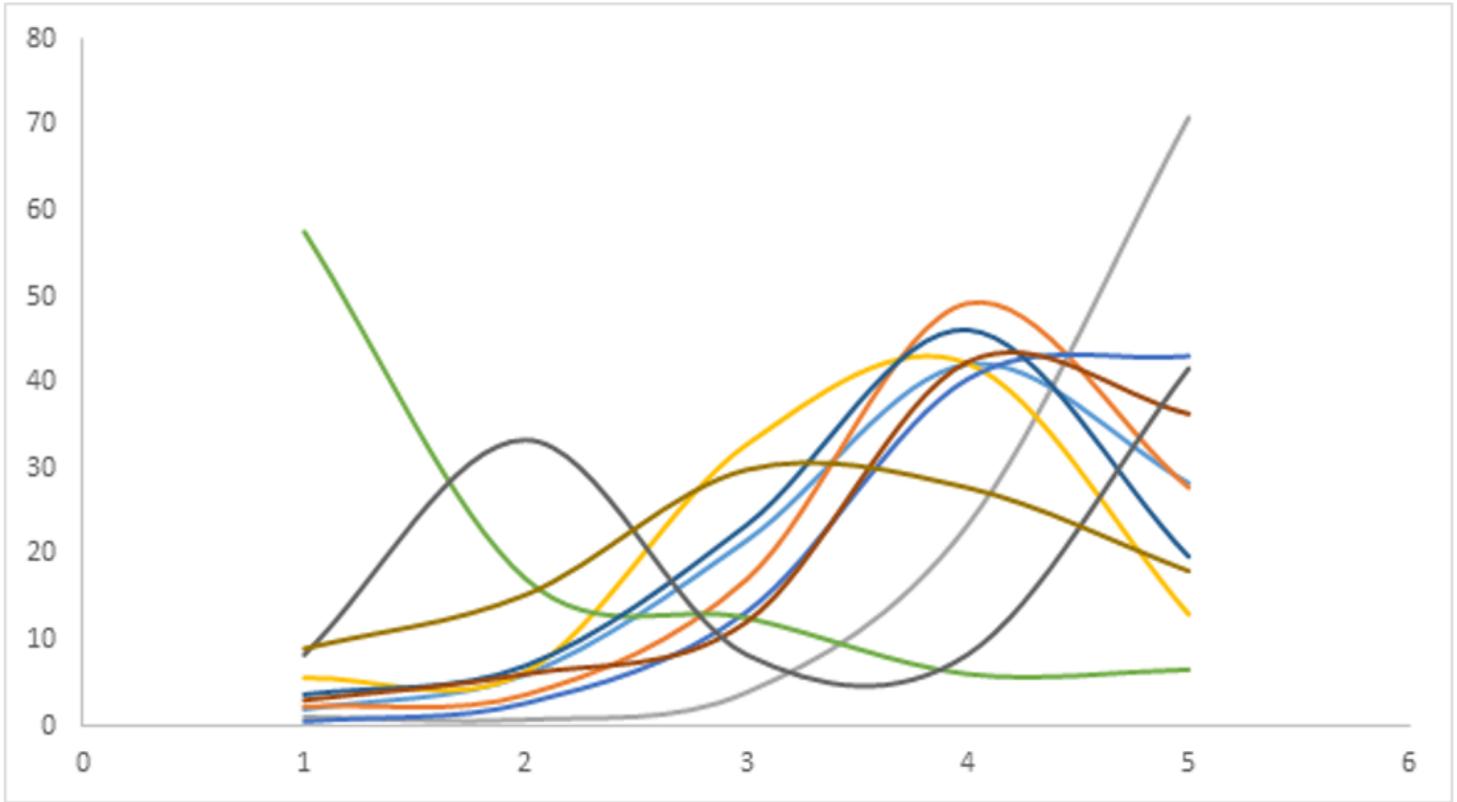


**Figure 8**

The correlation between “absolute quality” and “relative quality” from the user point of view for some messenger applications based on the IR-QUMA data. Every point represents average values of one data segment.

**Figure 9**

The correlation between “correctness” and “absolute quality” from the user point of view for some messenger applications based on the IR-QUMA Data. Every point represents average values of one data segment.



**Figure 10**

The histogram analysis of quality score levels (obtained from user judgments) about “absolute quality” for 10 different messenger applications. Every messenger application has five data points for the numeration of 1-very poor, 2-poor, 3-moderate, 4- good and 5-excellent quality scores. These scores reflect the user experience point of view to the quality of messengers.

**Figure 11**

The structure of a user opinion record

**Figure 12**

Error reduction curves for five different experiments and their average

### Figure 13

The optimization performance of the fitting-solving-procedure in the execution of the DD-KARB benchmark in Experiment-1 (Telegram). Every point represents the error of one state in the total state space. The fitting-solving-procedure outperformed the random baseline and found a state near the exhaustive minima.