



University of
Zurich^{UZH}

ARVV: Automated Refinement, Validation, and Verification of Financial Contracts

Simon Hurwitz, Dmytro Polyanskyy
Zurich, Switzerland
Student ID: 14-740-237, 16-909-160

Supervisor: Bruno Rodriguez, Eder Scheid
Date of Submission: June 2020

German Summary

Einleitung

Die Einführung von FINIG und FIDLEG bedeutet für Vermögensverwalter, dass Ihre Kundenverträge und die damit verbunden Anlagerichtlinien vollständig überarbeitet werden müssen. Zusätzlich sind Vermögensverwalter neu gesetzlich verpflichtet, ein tägliches Investment Controlling vorzunehmen. Da ein effizientes Controlling nur anhand von hochspezifischen Compliance Systemen erfolgen kann, müssen die Anlagerichtlinien erst nach den neuen gesetzlichen Anforderungen erstellt und anschliessend in die Systeme codiert werden. Bisher geschieht dies immer noch auf manueller Basis. Ebenso gefordert sind die Regulierungsbehörden. Je unterschiedlicher und komplizierter die Anlagerichtlinien, um so schwieriger gestaltet sich eine faire und einheitliche jährliche Prüfung im Rahmen eines Audits. Die Etablierung eines Standards ist deshalb von zentraler Bedeutung

Ziele

Das Ziel dieser Arbeit ist es, die standardisierte Erstellung von Vermögensverwaltungsverträgen und Anlagerichtlinien bei gleichzeitig automatisierter Validierung und Verifizierung dieser anhand eines Proof of Concepts (PoC) aufzuzeigen. Das PoC sollte dabei den zentralen Anforderungskriterien der Zeitersparnis und Reduktion von Komplexität sowie manueller Bedienung Rechnung tragen können. Um die Funktionsfähigkeit des PoC gewährleisten zu können musste beim Design die gesamte Wertschöpfungskette berücksichtigt werden: Von Attributen, durch welche sich ein Finanzinstrument charakterisieren lässt, über die Wahl von geeigneten Programmiersprachen bis hin zur Generierung von rechtlich vollständigen und korrekten Anlagerichtlinien.

Resultate

Unser PoC zeigt, dass Vermögensverwaltungsverträge zu standardisieren und gleichzeitig zu prüfen ob die Anlagerichtlinien korrekt erstellt wurden. Dies gelingt uns, indem wir Robotics Processing Automation (RPA) und Controlled Languages (CL) einsetzen, um den Vertrag zu validieren und in der Überprüfungsphase auch die entsprechenden Texterhalte zu generieren. Ferner zeigen wir, dass unser Tool die Vermögensverwaltungsverträge

inklusive der Anlagerichtlinien automatisch verfeinert, validiert und verifiziert. Dies trägt zu einer enormen Effizienzsteigerung bei gleichzeitiger Minimierung eines Fehlerpotentials bei. Die praxistaugliche Bewertung unseres PoC wird durch eine Usability-Studie und einen Anwendungsfall unterstützt, welcher die Vorteile der Verwendung unseres Modells gegenüber dem papierbasierten Status-Quo hervorhebt.

Weitere Arbeiten

Unsere Implementierung könnte sich konkret auf zwei verschiedene Gebiete ausweiten. Zum einen liessen sich mit unserem Tool jegliche Formen von Standardverträgen erstellen. Zum anderen könnten anhand von Machine Learning Verfahren die frühzeitige Erkennung von Verstössen gegen Vermögensverwaltungsverträge und Anlagerichtlinien ermittelt werden.

Abstract

In Switzerland, asset managers have a responsibility to adhere to financial regulations and guidelines under the Collective Investment Schemes Act (CISA) when it comes to managing their clients' portfolios. Although they do have a certain range of free discretion when it comes to making investment decisions, they are still contractually obligated to stick within certain bounds as per the aforementioned guidelines. Today, checking contracts for compliance with these investment guidelines requires a high manual effort in both its collection and enforcement stages. Our project endeavours to showcase that implementing a tool which enforces these contractual obligations in the active management of a clients assets is possible and increases both accuracy and efficiency. In other words, we build a Proof of Concept (PoC) that reduces the risk of both an active and/or a passive violation of investment guidelines through the Automatic Refinement, Validation, and Verification (ARVV) of these financial contracts.

Acknowledgments

We would like to express of thank to both supervisors, Bruno Bastos Rodrigues and Eder John Scheid for their continuous support, guidance, feedback, valuable input and related work throughout the whole development of this Master Project. Furthermore, we would also like to thank the Swiss Stock Exchange (SIX) for their supportive data contributions which made our proof-of-concept possible. We would also like to express our gratitude towards the Communication Systems Group and Prof. Dr. Burkhard Stiller for the opportunity to conduct this Master Project.

Contents

Abstract	iii
Acknowledgments	v
1 Introduction	1
1.1 Motivation and Problem Description	2
1.2 Goals	2
1.3 Structure	3
2 Related Work	5
2.1 Controlled Language Processing	5
2.2 Robotics Processing Automation (RPA)	6
2.3 JSON Format	8
2.4 Data Classification	8
2.5 Software Testing and Usability Studies	10
2.5.1 Usability	11
3 Investment Controlling Elements	15
3.1 Regulations	15
3.1.1 Permitted Investments	15
3.1.2 Limitations	16
3.1.3 Investment Techniques	16
3.1.4 Prohibited Systems	16

3.1.5	Risk Diversification	16
3.1.6	Further Limitation	17
3.2	Contract Design	17
3.2.1	Rule Construction (Rule Structure)	17
3.3	Contract Implementation	23
3.3.1	Impact Assessment of Contract Content	24
3.3.2	Check Completeness of Data	25
3.3.3	Collection, Preparation and Use of Data	26
3.3.4	Transformation of Contracts into Rules	26
3.4	Investment Guideline Monitoring	28
4	Design	29
4.1	Data Refiner	29
4.2	Data Validator	30
4.3	Data Serializer/Deserializer	31
4.4	Data Verifier	32
4.4.1	Flag Value	32
4.4.2	Examples of Rulesets	32
5	Implementation	35
5.1	Data Standardization and Scrubbing	35
5.1.1	Data Scrubbing and Refinement	35
5.1.2	Database	37
5.1.3	Form Builder	38
5.1.4	JSON Schema	38
5.2	Contract Verification and Generation	39
5.2.1	Controlled Language Processing	39
5.2.2	Python RPA	40
5.2.3	Contract Manipulation	41

<i>CONTENTS</i>	ix
6 Evaluation	43
6.1 Effectiveness, Efficiency and Accuracy	43
6.1.1 Error Reduction	43
6.1.2 High Uniformity and Precision	44
6.1.3 Time and Cost Savings	44
6.2 Usability Study	44
6.2.1 Background of Usability Study	45
6.2.2 Further Features	45
6.3 Use Case	46
6.4 Challenges and Future Work	47
7 Conclusion	49
Abbreviations	51
List of Figures	51
List of Tables	53
A Contents of the CD and Installation Guidelines	57
A.1 Angular Form	57
A.2 Python Verification	57
A.3 Contents of the CD	58
B Usability Study Results - PSSUQ	59

Chapter 1

Introduction

Laws and normative regulations are the basis of legal contracts formalized with clients of financial service providers, *i.e.* banks. Recently, these contracts began requiring a check for compliance on a regular basis, which is usually provided by an external service provider, *i.e.* legal consultancy company. Asset managers, who have to track their clients' investment strategies daily, and trade in the financial markets, are particularly affected by these regulations and the associated costs. Thus, a salient objective of an asset manager's daily compliance control is to immediately report potential and actual breaches of a contract to key stakeholders, as well as begin taking steps to correct the breach [23]. In this sense, the goal of the asset manager is to ensure the integrity of the contract, reporting any deviations from the paper-based guidelines. This forms the core of investment controlling, which is located within operational risk management. The implementation of an efficient investment controlling process requires high-performance software tools with various interfaces to different portfolio management systems, data providers, and custodian banks. From the point of view of an investment controller, an ideal mandate life cycle consists of the following components:

- Regulatory
- Contract Design and Implementation
- Investment Monitoring and Reporting

The Contract Design and Implementation stage of contract's life-cycle is especially important for daily investment monitoring as it has a direct impact on investment's monitoring and reporting [30]. The latter can only be efficiently conducted if the contracts are designed in such a way that the investment controlling software has the ability and options to check the entire content of a contract. Furthermore, it is essential to translate high-level contract terms into rules defined within a investment controlling software.

Due to detailed requirements of financial market authorities, standard contract templates are highly demanded to avoid potential conflicts in the contract's design and implementation. This is true for both small and large asset managers. Thus, standardized translation

rules for the refinement of a contract into verifiable code should exist for investment controllers [13]. Hence, based on the growing digitization of the asset management industry, standardized software solutions to remedy the above problems are of great economic interest.

1.1 Motivation and Problem Description

According to MiFID II [8] in Europe and FINIG FIDLEG [7] in Switzerland, an asset manager is required to have the implementation of the contracts with his or her clients, and the related suitability and appropriateness checks verified by a third party on a pre- and post-trade basis. Thus, even in large companies, investment guidelines and contracts require high manual effort in its both the design and implementation stages. After contracts have been created with the customer, these restrictions are manually coded in compliance systems. This has the consequence that neither a uniform contract template or settlement nor a uniform coding is possible. Furthermore, any contractually agreed upon restriction must be translated into an SQL code and subsequently built into the compliance system. The potential for errors is enormous because only when coding the restriction it becomes apparent whether it can be actually be registered in the system accurately and in its entirety. If not, the contract and investment guidelines must be reformulated and returned to the client. This carries an inherent efficiency problem as well as the potential for a firm's reputational damage in the long term. Therefore, the goal of this master project by means of a Proof of Concept (PoC) is to combine and automate the creation of a standardized contract and its verification with corresponding investment guidelines.

1.2 Goals

As outlined in the previous section, the main goal of this Master Project is to design and implement a PoC that combines and automates the creation of contracts that are verified with corresponding investment guidelines. In this regard, the production of software as a result of the contract's refinement should, simultaneously, facilitate several tasks for the financial service provider and the financial market supervisory authority. Driven by the description of the outlined work, the following master project goals are required:

- Standardization of asset management contracts and investment guidelines based on a pre-defined questionnaire: Depending on the involved complexity and provider the creation of asset management contracts and investment guidelines usually costs between 4000 - 5000CHF per contract and takes between 1-2 days to create them. Unlike the contracts, there are no standards for creating investment guidelines today. Asset managers can save time and money by standardizing investment guidelines based on a pre-defined questionnaire. Behind each question on the questionnaire, a text preserve is deposited. After completion of the questionnaire, the asset manager receives a string of text preserves, which together form the investment guidelines.

Financial market supervisors can concentrate on only one format during the year-end review. This increases the efficiency of financial market supervision, which in turn can reduce costs for the asset manager and thus for the entire market. In addition, the potential for errors and thus the operational risk are minimized.

- Translation of asset management contracts and investment guidelines into JSON/SQL codes and embedding them in portfolio management and compliance systems: After creating the asset management contracts and investment guidelines, these must be translated into a code and maintained in an Investment Controlling software. The tools (Simcorp, MIG21, Allocare, Sentinel, Aladdin, Charles River and Bloomberg Compliance) commonly used in the market are all based exclusively on SQL. Due to the lack of standard investment guidelines, this is currently done individually for each portfolio. This means that each portfolio is individually coded into the investment controlling software. Due to the standardization of the investment guidelines, it is possible to assign an SQL code to every text preserves described above. This ensures that the standardized guidelines must be coded exactly once and not for each individual portfolio as it is the case today. This results in a large savings and efficiency potential. As described above, nearly all investment controlling products available on the market are based on a SQL basis. The software to be created in this master project would therefore be easy to embed in any SQL software system.

1.3 Structure

This Master Project will be organized by first presenting a few key concepts to the reader in the following *Related Work* chapter which are central to our project and subsequent proof-of-concept (PoC). Namely, Controlled Language Processing, Robotics Processing Automation as well as background on a few technologies and methods which we refer to at later sections as a basis for our work. In the ensuing chapter, we will present key information regarding investment controlling elements so the reader has the appropriate background to understand the PoC. In the subsequent two chapters, we will then explain the Design of our PoC, followed by a detailed technical explanation of how we implemented it, respectively. Finally we will endeavour to present an Evaluation of our PoC through a usability study and a use-case which highlights the advantages of using our model over the paper-based status-quo.

Chapter 2

Related Work

In this section we endeavour to showcase some related work which are key for our ARVV PoC tool and can serve as a background to the reader. Firstly, we delve into Controlled Natural Language (CNL) - as this is a major basis for creating a standardized contract. Secondly, Robotics Processing Automation (RPA) is central to the subsequent validation and verification of our financial contract. Lastly, we also take the time to go over some data classification schemes for investment controlling as well as software and usability studies which serve as the basis for our Chapter 6 Evaluation section.

2.1 Controlled Language Processing

As a subset of Natural Language Processing (NLP), Controlled Natural Languages (CNL's) or Controlled Languages (CL's) are a salient tool for us when it comes to developing our PoC as it allows us to formally restrict the Financial Contract to a subset of clauses, phrases and keywords. In this section, we will discuss some definitions and uses of CNL's.

There are many instances where CNL's are employed in order to reduce ambiguity while simultaneously presenting simplicity as a trait. CNL's are often an umbrella term for languages that are able to improve communication among humans, to improve translation, or to provide natural and intuitive representations for formal notations [20]. CNL's can have many different attributes - ranging from: controlled, processable, simplified, technical, structured and/or basic [20]. Such a wide definition often results in no agreement being reached on what a true CNL really encompasses. Today, it is commonly accepted that CNL's exist in many environments - namely, computer science, philosophy, linguistics, academic and government.

Kuhn [20] says that CNL's can also sometimes be intentionally ambiguous while holding different formats - for example, some many look very similar to English, some many be defined by a few grammar rules, while others may look more like a programming language.

CNLs appear to be particularly useful in respect to information extraction of and reasoning with the content of documents on the internet [32]. An interesting example of a CNL is SBVR Structured English. It is a CNL for business rules first presented in the mid-2000s. According to Kuhn [20], the vocabulary consists of four types of sentence constituents: terms (i.e., concepts), names (i.e., individuals), verbs (i.e., relations), and keywords (e.g., fixed phrases, quantifiers, and determiners). The following is an example:

A rental *must be guaranteed by* a credit card *before* a car *is assigned to the* rental.
Rentals by Booking Mode *contains the categories* 'advance rental' *and* 'walk-in rental'.

Figure 2.1: SBVR Business Language

We realize that for our implementation, a CNL similar to SBVR was very feasible to develop as financial contract law oftentimes has repetitive fixed phrases, quantifiers, determiners - with primarily only names, amounts and quantifiers changing.

2.2 Robotics Processing Automation (RPA)

Robotic Process Automation (RPA) describes the technological replacement of human employees. The main goal is to perform structured tasks more efficiently than the ones executed by humans. RPA usually means nothing more than a finite automaton that mimics a person's possible activities. Every possible state is mapped in the software and depends on the previous decision, which can also be influenced by humans as well [10]. RPA is integrated into IT systems via a front end. In practice, this means that the software robot uses IT systems just like a human, repeats precise steps and responds to events on a computer screen instead of communicating with the system's API [16]; RPA is able to process entire activity chains in a logical link. Tasks are performed as if a real person was doing tasks across systems. RPA robots thus combine the following properties: [16]

- They act as virtual workers that are monitored by people. Automated processes can easily be changed by software robots even by users of the system. Traditional software requires advanced coding skills to make major changes in how it works.
- They take over the automation of repetitive tasks of humans. On the other hand, software robots can be instructed by changing relatively simple logical tasks, capturing the process performed by a human on the screen. This makes RPA very versatile and flexible. Therefore RPA can be implemented in a very short time.
- They are integrated into the existing IT architecture and do not require complex interfaces. Many corporate IT systems are proprietary and have no public APIs, which severely limits their ability to communicate with other systems. Therefore RPA can be implemented in a very short time.

The crucial point is that RPA operates with its own user name and password in existing IT landscapes without cumbersome adjustments. The software-controlled coordination of activities works across different applications, for example in existing compliance systems [29]. Nowadays, many companies use RPA to automate structured tasks. A link between research and business has also developed, which is an interesting example of the Robolaw project. This is funded by the European Commission under the 7th Framework Program for Research and Technological Development (FP7). The aim of the project was to use a mixture of research and non public advisory services to investigate the emerging technology, whether the existing laws in Europe can be simplified or rendered obsolete with this technology and whether regulations should be developed instead or not [17].

In our ARVV project, the precise handling of individual activities to create contracts and investment guidelines is of central importance. Both fund contracts with integrated investment guidelines and investment guidelines as part of asset management contracts can be represented. The overarching goal of our work was to develop a tool with which we could deal with the documents mentioned in the most efficient way possible. The challenge was as follows: The creation of legally valid contracts and investment guidelines as part of the law, taking into account a practical use within the field of asset and data management. On one hand, the contracts must take the legal system into account, while on the other hand, the investment restrictions contained therein must under no circumstances contradict each other. In addition, the investment restrictions must be formulated and constructed in such a way that they can also be represented in compliance systems and that the required data to perform guideline checks is also available for the individual guidelines, asset classes and instruments. During our research, we found that this interdisciplinary area was not sufficiently covered by academia, but the use of RPA in the respective individual areas was used. The use of RPA in the field of law is of particular importance for this work.

Remus, D et al. estimated the average of the working hours of different work steps of a lawyer when dealing with law cases [26]. The time usage data comes from Consilios Sky Analytics of Framingham, Massachusetts, a consulting firm that provides corporate clients with the aggregation and analysis of invoices billed by law firms. Each invoice covers a small period of time and describes the work step carried out by the lawyer using a task code from the ABA's Uniform Task-Based Management System (UTBMS). The UTBMS consists of 114 different task codes, which are summarized in 13 categories in Figure 2.2.

The employment impact is maximum moderate and it quickly becomes clear that human effort can be replaced pretty easily. The requirements that a system must have in order to be marketable requires a combination of legal rules and heuristics, which are used in an argumentation machine as if-then statements, decision trees, mathematical formulas as well as weighing tests and formalized directly in a code. These relationships provide the structure of the data on which the system works [27]. When designing our tool, we made sure that we use closed questions or options at the front end to eliminate possible overlaps. At the back end, the results of the system, *i.e.* investment guidelines, can be presented as customized documents.

Task	Tier One Firms	Tier Two-Five Firms	Employment Impact
Document Management	0.4	0.7	Light
Case Administration and Management	3.7	5.6	Moderate
Document Review	4.1	3.6	Strong
Due Diligence	2	3.4	Moderate
Document Drafting	5	4	Moderate
Legal Writing	11.4	17.7	Weak
Legal Research	0.5	0.4	Moderate
Legal Analysis and Strategy	28.5	27	Moderate
Fact Investigation	9.2	9.6	Weak
Advising Clients	9.3	3.2	Weak
Negotiation	3	5	Weak
Other Communications/Interactions	8.8	5	Weak
Court Appearances and Preparation	13.9	14.5	Weak
Totals	99.8	99.7	

Figure 2.2: Employment Impact of Document review

2.3 JSON Format

JavaScript object notation (JSON) is a plain text data exchange format. JSON is the leaner alternative to XML, since the same functions can usually be available except with more simple analysis measures. JSON can be used very flexibly, offering high throughput and low latency without sacrificing scalability [25]. The format is also supported in JavaScript and is therefore suitable for HTML-based representations. Thus, JSON offers a more appropriate performance compared to XML and other relational databases [12] [33].

The use of JSON as a new approach to processing and managing large amounts of data is relatively new in research. The simple data exchange format is particularly suitable for standardized processes such as the creation of investment guidelines. It is essential that it is easy for people to read and easy for machines to analyze and generate [21] [10]. Various results show that the JSON approach to query storage and retrieval is more powerful and efficient than XML. When scalability is of importance, this format that can process large amounts of data quickly and without losses. This is particularly important in applications where many users access a tool at the same time to conduct operations [15]. Furthermore, JSON can be easily translated into SQL queries for proprietary investment controlling software.

2.4 Data Classification

With regards to the formal classification of securities, various problems have existed in practice for decades. These are mainly [28]:

- The lack of an industry-wide or internationally accepted approach for the classification of financial instruments.
- The use of similar terminology for instruments with different characteristics.
- This does not allow securities to be grouped in a consistent manner, resulting in different categorisations of reports on holdings.

The advantages of a uniform code are:

- Definition and description of an internationally valid system for the classification of financial instruments
- The provision of codes that can be used by all market participants in their environment, which would lead to an increase in the quality of information exchange

Classification techniques are becoming increasingly important in the financial world to reduce risks [11]. Managers are not only interested in high accuracy, but also in interpretability and transparency and it is now generally accepted that understanding the relationship between inputs and outputs is crucial for operational and strategic decisions [14]. There are a wide range of classification standards in place, but none of them have been widely adopted to date. The reason for this is clear: there are thousands of asset managers who have concluded contracts with their clients and setting up a standard in an already established environment brings only indirect advantages for the individual asset manager..

For an efficient analysis of financial data, they must be broken down into blocks using a top-down approach. This allows data noise and redundancy to be eliminated so that a classification standard can be created. The choice of the method that contributes to the creation of a framework is crucial. First we investigate the classification standard according to ISO 10962 [6]. ISO 10962 defines the structure and format for the classification of financial instruments approved by the International Organization for Standardization (ISO) These instruments are generally organized in groups called *Asset Classifications*. The ISO classification scheme has six different asset categories:

- Equities
- Debt instruments
- Rights
- Options
- Futures
- Other

The major flaw of this categorization is that the largest part of this classifications scheme is the loosely defined Other. Another issue is that Equities and Debt instruments are instruments as well as asset categories whereas Options are only instruments and not categories - which leads to confusion [6].

Another classification scheme is the one from the World Bank 2008 SNA [9]. This classification scheme has eight different asset classifications and shares some similar issues as the one presented above:

- Monetary gold and special drawing rights
- Currency and deposits
- Debt securities
- Loans
- Equity and investment fund shares
- Insurance, pension, and standardized guarantee schemes
- Financial derivatives and employee stock options
- Other accounts receivables/payables

We recognize the problems and issues surrounding the various classification standards as well as the confusion between asset categories and financial instruments in certain cases. As a result, we take this into account when designing our ARVV tool.

2.5 Software Testing and Usability Studies

For many companies, common test types are functional tests, compatibility tests, performance testing, scalability testing, usability testing, application security testing, accessibility testing and regulatory compliance testing. For our purposes, we primarily focus on usability testing and a brief background will be provided below.

Today's software systems have generally been subjected to extensive human factor studies. Nevertheless, an analysis of human factors is still a very subjective matter. There are various points that need to be examined in a usability test [22]:

- The user interface is tailored to the intelligence and educational background of the end user
- Relevance of the program's expenditure
- Uncomplicated and easily understandable error diagnosis and error messages

- Consistency and uniformity of user interface syntax, conventions, semantics, format, style and abbreviations
- Existence of sufficient redundancy
- Omission of unnecessary options and features
- Receipt of confirmations for entries
- User-friendliness
- Usability
- Design of the user interface is adapted to the user
- Safety when navigating between different options
- Fulfilment of the design promise

If a user determines that a particular application does not conform to the specifications due to an incorrect design, a cumbersome user interface, or missing or ignored specifications, the development process has failed [18].

2.5.1 Usability

Interestingly, strong and accurate results in usability studies are achieved with as little as 5 users comprising the test. Tom Landauer has shown that in a usability test with n users, the following number of usability problems were found: $N(1 - (1 - L)^n)$ where N is the total number of usability problems in the design and L is the percentage of usability problems detected during testing of a single user [24]. Landauer finds that the typical value of L is 31% per cent, averaged over a large number of projects that have been investigated. Furthermore, as soon as data is collected from a single test user, the findings increase and almost one third of all information about the usability of the design has already been accounted for. The study finds that the difference between zero data and just a small amount of data is quite significant.

For instance, as the second user is tested, the person performs some of the same actions as the first user, so there is some overlap in what is derived from the study. Landauer finds that even though people are definitely similar, there will most likely be something new that the second user found which the first did not. In other words, the second user will add some new insights, but not nearly as much as the first user.

The third user will do many things that were already observed with the first user or with the second user, and even some things that were already seen twice. The third user will generate an even smaller amount of new data. The curve of additional information is first steeply rising and then declines more and more flatly. In order to be able to identify all usability problems in the design, at least 15 attempts are needed, which would ideally be divided into three studies with 5 users each.

Usability Questionnaire

As with the software test procedure itself, a usability questionnaire should be carefully planned to obtain the information required from the associated test procedure. It is important to leave out free response texts in order to get the best possible comparability of the answers. One of the most frequently used and quoted questionnaires with over 2,000 citations is the Post Study System Usability Questionnaire (PSSUQ).

Description of the PSSUQ

The PSSUQ is a questionnaire to assess the perceived satisfaction of users with computer systems or applications. The origin of the PSSUQ was an internal IBM project called SUMS (System Usability) MetricS), led by Suzanne Henry. The SUMS researchers created a large pool of elements based on the contextual usability work of Whiteside et al (1988) [31].

The PSSUQ elements generate four ratings - one overall and three subscales. The rules for the calculations are:

- Total: average of the answers for points 1 to 16 (all points)
- System quality (SysQual): Average elements 1 to 6
- Information quality (InfoQual): average points 7 to 12
- Interface quality (IntQual): Average elements 13 to 15

We leave an example of PSSUQ for the reader on the next page (Figure 2.3).

Question Nr.	The Post-Study Usability Questionnaire Version 3	Strongly agree	1	2	3	4	5	6	7	Strongly disagree	NA
1	Overall, I am satisfied with how easy it is to use this system.										
2	It was simple to use this system.										
3	I was able to complete the tasks and scenarios quickly using this system.										
4	I felt comfortable using this system.										
5	It was easy to learn to use this system.										
6	I believe I could become productive quickly using this system.										
7	The system gave error messages that clearly told me how to fix problems.										
8	Whenever I made a mistake using the system, I could recover easily and quickly.										
9	The information (such as online help, on-screen messages and other (documentation) provided with this system was clear.										
10	It was easy to find the information I needed.										
11	The information was effective in helping me complete the tasks and scenarios.										
12	The organization of information on the system screens was clear.										
13	The interface* of this system was pleasant.										
14	I liked using the interface of this system.										
15	This system has all the functions and capabilities I expect it to have.										
16	Overall, I am satisfied with this system.										

Figure 2.3: Usability Questionnaire

Chapter 3

Investment Controlling Elements

3.1 Regulatories

The density of regulations on the European and Swiss financial markets has increased significantly - and is increasing in parallel with the progressing globalization of these economies. Often, market access to new and expanding areas can only be guaranteed through agreements and rules. Companies that are especially active in several business sectors are faced with major challenges since they often have to comply with various laws and requirements at the same time. In the meantime, the fulfillment of regulatory requirements is a major cost driver. In order to continue to be successful in this dynamic environment, holistic impact assessments, efficient interpretation of regulations and the definition of business-oriented frameworks are necessary. The following investments restrictions can be found in the CISA Art. 53 ff [5] and must be integrated in every asset management contract of an investment fund. We therefore used several FINMA Compliant fund contracts from large banks as a basis to cross check, whether or not we have included all the required restrictions. As our work focuses on asset managers as a whole, we have integrated this high level standard into our solution discussed in further sections. As there is no direct research regarding Investment Controlling, we set up a clear framework for the theoretical concepts behind state of the art investment controlling. In the following section we will define different requirements that we envision for our ARVV tool.

3.1.1 Permitted Investments

- Cash and cash equivalents (bank deposits, deposits / time) "to a limited extent"
- Derivative financial instruments max. 210 percent total engagement
- Shares in collective investment (from 85 percent feeder fund)
- Money market instruments
- Effects
- Other plant max. 10 percent

3.1.2 Limitations

- Derivatives: Underlying permitted in accordance with fund regulations and trading on the stock exchange; OTC derivatives possible, provided they can be traded daily or returned daily.
- The same provisions regarding investment policy etc. apply to target funds and the target fund is adequately monitored.
- Money market instruments: liquid and assessable as well as trading on the stock exchange or issuance of SNB, EU central bank, central bank EU member state, etc.

3.1.3 Investment Techniques

- Direct or indirect systems (without leverage)
- Securities lending and repurchase agreements (repo, reverse repo) permitted with regard to efficient management of the fund's assets (prerequisites according to Art. 1-22 CISO-FINMA)
- Borrowing (max. 10 percent) and pledging / granting collateral (max. 25 percent)

3.1.4 Prohibited Systems

- Precious metals, precious metal certificates, goods, commodity papers
- Short sales
- Lending / providing guarantees

3.1.5 Risk Diversification

- Securities or money market instruments of an issuer max. 10 percent
- Total investment in issuers > 5 percent max. 40 percent
- Current and temporary credit (including liquid funds) at a bank max. 20 percent
- OTC transactions with a counterparty (if not secured) max. 5
- OTC transactions with bank as counterparty (if not secured) max. 10 percent
- Investments, balances and claims against an issuer max. 20 percent
- If publicly guaranteed or issued max. 35 percent
- Shares in other collective investments max. 20 percent
- Shares in non-UCITS / CH securities funds max. 30 percent

3.1.6 Further Limitation

Limitation of participation in a single issuer

- Participation rights of over 10 percent in an issuer
- Non-voting equity securities in an issuer of more than 10 percent
- More than 25 percent of the shares in another KKA

3.2 Contract Design

Asset managers are obliged to work out the appropriate investment strategy together with their clients and if necessary, with fund companies. They must also subsequently document this. All future asset management is based on the investment strategy which provides a legally binding basis in conjunction with enforceable guidelines. However, an investment strategy simply defined in writing can often lead to different interpretations and therefore represents a great potential for misinterpretations and therefore a great risk of legal disputes for both parties. As a result, it is important to describe the strategy that is to be pursued as clearly as possible. It is therefore advisable to draw up contracts and the associated investment strategy and risk information carefully with the customer. From today's perspective, an asset manager should be able to operate within the limits set out in the contract at their own discretion. If the definition of free discretion is unclear, there is also a high risk present. For this reason, this free discretion of the asset manager should be contractually stipulated, on the one hand to reduce sufficient flexibility in the active management of client assets, and on the other hand to reduce the risk of an active or passive violation of the investment guidelines.

3.2.1 Rule Construction (Rule Structure)

The following explanations are based on the industry standard.

Control the rule - Test process type

If one checks the investment guidelines on a simulation model of positions for possible violations, one speaks of a pre-trade check. The portfolio manager uses this as standard and uses current trading prices ("intraday") and current position positions, which takes into account any trades made. The test results must be calculated quickly and do not correspond to the results of the mandatory post-trade control. This takes the reconciled end-of-day stock and the closing prices on the trading day. The check is usually carried out on the next trading day or at the end of the inspection and reporting period. It must take place at least as often as the net asset calculation. The portfolio manager must not carry out the follow-up inspection for the customer and the results must be archived. This serves as evidence of correct management. Furthermore, violations must be forwarded to

the legal department in order to initiate any corrective measures. In addition, the violations must be reported to the customer.

Technically speaking, the checks on model scenarios (pre-trade) are triggered ad-hoc, while the control of contract compliance is carried out automatically after the end of the day processing. In the case of ad hoc tests, database queries are triggered, which carry out calculations according to the coded rules and compare the results with the specifications - these test results are displayed to the user. In bulk tests - as is typically the case in test cycles in post-trade monitoring - a sequence or time-controlled batch process usually triggers the test routines. The test results are often output in a report after a short manual check for verification.

A monitoring system can be easily built by a technically savvy person. The crux of the matter is not the rule coding, but the process flow during the test. This know-how must also be ensured in the long term. If you do not see this as a core competence, outsource the monitoring.

Assignment to mandate

An investment policy is checked using a procedure that uses position, transaction, instrument, reference and market data to determine the value of the target. These procedures are called rules and they are structured in such a way that the portfolio and key date or inspection period are variable. Accordingly, it is necessary to assign a rule to a mandate. Of course, this rule can also be assigned to another portfolio. The assignment is strictly for a period of time. In some systems this can be defined explicitly while in others there is a status attribute for activation.

Review period or key date

Depending on the investment guidelines, one looks at data from the portfolio that is only valid on one day or one takes data records from a period. Typical periods are months, quarters or years. The end of the month, quarterly or year-end are also suitable as reference dates. In this context, it is important to differentiate between how often checks are carried out. The test frequency is related to the cut-off date and test period in that a complete one.

Selection of Accounting Data

The booking data is selected using a database query, provided that the accounting is done electronically. Appropriate criteria are used to select the relevant, mandate-specific position or transaction data for a specific rule. These filters are based on the principle of set theory and can be found in the *where* clause of the database query. The data attributes relevant to the rule and any calculations are listed in the *Select* part of the data query. The types of data considered (positions, transactions, master data) are defined and linked in the *from part*.

We often also work with market value weights. The value of the reference - the total market value (NAV) - must first be determined. The master data is necessary to map the business processes. Master data on instruments held and associated valuations (market data) are required for the review of most investment guidelines.

Selection of Reference and Market Data

The two data types relate to the instructions of plant positions. In contrast to master data, it is over time and must be queried periodically. These values represent a time series for a personal data attribute or a data attribute tuple in which each tuple belongs to a defined sentence, such as a three-level sector classification. Typically, this data is used by service providers and when creating a contract, one must make a special purchase as to which Market Data System (MDS) is used.

Introduction to Set Theory

The basis for investment guidelines - and the database query language SQL - is set theory, which is why we will briefly discuss set theory here. Set theory knows a mathematical description of the linking operations. The quantity description is used for the selection of the considered investments.

Basic set (universe), set, empty (basic) set

The reference is a basic set - the whole - or a defined set. If it contains no element, one speaks of an empty set. The basic set consists of a defined number of elements: $G = \{x_1, x_2, x_3, \dots\}$. The empty set has no element $L = \emptyset$, but the universe comprises all elements. Examples of this are (a) investment universe, (b) shares or options on commodities and (c) bonds in a share fund.

Subset

Investment guidelines specify how much a subset of the basic amount may comprise. The subset comprises part of the elements of the basic set: $T \subseteq G$.

Examples include (a) German equities, (b) EU and Swiss bonds, (c) Money market bonds that mature in five or more years.

Union set

The union set consists of several partly overlapping/disjoint sets or from parts of several quantities. Intersections have common elements: $M \cap N$. Non-overlapping sets can be summarized as follows: $M \cup N \cup P$. In combination, non-overlapping subsets can be defined as follows: $(T \subseteq M) \cup (T \subseteq N)$. Examples include (a) equities and bonds, (b) leverage, yield and capital protection products.

Exclusion amount

The exclusion set is the elements of a defined subset or with certain values of a property or several subsets of a set that are excluded by ranges of values of different characteristic properties. Sometimes it is practical to make a quantity description using a basic set and to exclude some elements: $M \setminus N$. Examples include (a) bonds without bonds during the year, (b) shares of companies without dual-use goods production and (c) securities of companies from non-cyclical industries

Quantity combinations

Different quantity combinations allow the exact formulation of the correct target quantity. Shares A, B bonds, 'Structured products' S, 'Bearer shares' J, 'Preference shares' V, 'Convertible bonds' W, 'Bonds CHF domestic' D, 'Foreign currency bonds' F, 'Obligation EUR' E, 'Obli GBP' G, 'Obli USD' U. Finally, the issuer is common to all investments. The goal is to create a description which is precise enough.

- $A \cup B$ where issuer = X
- $J \cup V \cup W \cup D \cup E \cup G \cup U$ where issuer = X
- $(J \cup U \cup W) \cup B$ with currency = CHF, EUR, GBP, USD where issuer = X

Scattering

In contrast to the description of the quantity, the description of the target value is based on the mean value and the variation.

Reference size

Comparisons between sets or based on sets yield one or more binary or numerical referential results. The reference quantity serves the most common test method. Absolute results should also be counted here. A comparison is described by means of lower and upper limits..

Deviation

If you examine deviations, you orientate yourself towards a target value to scatter the results of the comparison. The result set is formed by mean and deviation.

Types of Rules

All types of rules can be applied to five basic test samples:

- A result is determined for the entire account.
- One result is determined for each position.
- One result is determined per transaction.
- A result is determined that relates to a step in the investment process.
- A result is determined that is collected over a test period.

The results are determined in different procedures which all correspond to a specific rule type:

Bandwidth rule

Often there are ranges or upper or lower limits for a certain subset of investments or assets. These minimum and maximum values can be defined absolutely or relatively - with reference to another (sub) quantity of the investments or assets or a defined reference quantity. Bandwidths consist of an upper and a lower limit. If only a limit is specified, this must be exceeded or fallen short of, depending on the type. A method developed for pre-trade control (pre-trade control) reports warnings in an area close to the limit values and the violation if the limit value is exceeded or fallen below. The size of the area can be determined, usually using a percentage (e.g. 98% of the value at a maximum value).

Target rule

In contrast to the bandwidth rule, a target value is targeted and an accepted symmetrical deviation is defined via a tolerance. If the aggregation of a certain subset of investments results in an amount that is within the described range, the test result is positive. Performance targets are monitored in this way, for example, but they often represent target values instead of restrictions.

Deviation rule

The deviation rule is related to the target value rule. An accepted symmetrical deviation is defined by means of tolerance. If the aggregation of a certain subset of items results in an amount within the range described, the test result is positive.

Projection rules

If monitoring based on non-numerical values is announced, a projection method is used. The discrete values are assigned numerical values in order to make a mathematical comparison. The mapping of the non-numerical values must be complete, but not disjoint. The projection function must represent an image in an orderly space with at least one boundary between the permitted and unauthorized range of values. Rating criteria are the most common application of the projection rule. Creditworthiness or counterparty risks are assessed using accounting decision-making processes with an abstract classification using a code of letters and characters. A few rating methods are used as standard in the market data offerings.

Approximation rule

If a contract passage cannot be checked due to the data available or if the rule is intended for investments that will later or probably never be carried out, a check rule is often coded. This rule reports the occurrence of the transaction or new position as a rule violation. Sometimes the calculations are so complex that a simplified check provides a basic result regarding rule violations. For example, an average bond duration of over a year in the absence of bonds can be excluded; the instrument classification almost always gives the clue here.

Manual rule

If systemic support is insufficient, investment guidelines - especially regulatory requirements must be evaluated manually based on the positions. This is often done in a spreadsheet. Sometimes auxiliary applications are also created for reasons of efficiency. These rules hold a hidden potential for errors among the primary reason being human error. Furthermore, the variety of test options and their complexity is significantly higher than with automated monitoring.

Complex Structures

Combination rule

Depending on the test object, it is necessary to connect self-contained test materials. The results of the individual test parts are linked according to the set theory. There are four set operators:

- *a* AND *b* The condition is fulfilled if event *a* and event *b* apply.
- EITHER *a* OR *b*: The condition is met if only one event applies.
- *a* OR *b*: The condition is met if at least one event applies.
- NOT *a* AND *b*: The condition is fulfilled if event *a* does not apply and event *b* applies.

Trigger rule

If the customer's consent to an investment (scenario) is required, the occurrence of an investment or the achievement of a threshold value (market data) triggers an extended / changed control. One saves the coding or activation of one or more extensive guidelines, so one is able to set a test rule as a trigger.

Linking Rules

The assignment of a portfolio to a rule is possible in systems by direct connection or by classifying a portfolio in a portfolio structure. In any case, the rule is assigned to a bucket in the hierarchy. Inheritance takes place downwards in multi-level hierarchies.

Types of Rules

The type of rule typifies transformed investment guidelines in terms of finance and craftsmanship. In addition to the test topic, the need for data can be concluded. There is no direct connection between rule types. Below is a sample list of some rule types:

Regulatory rules

Jurisdictions regulate the financial industry via laws called regulations. These are regulations on investor protection and financial (market) stability in the legal area. The transformation of the guidelines are the regulatory rules. They form libraries and must - at least in part be applied to all mandates.

Benchmark rules

Common to this type of rule is its reference to indices or benchmarks. Investment guidelines agreed with the client refer to a possibly composed benchmark.

Sustainability rules

Test rules check the investment guidelines of sustainable investments. Sustainability is defined as Portfolio management specific rules. These test methods are used for process support in investment management and do not represent contractual agreements or regulatory requirements. Naturally, they are used in scenario modeling in pre-trade control and are rarely used to monitor the portfolio manager.

Concentration rules

These checking rules calculate shares and thus control compliance with the investment profile of a mandate. They relate to a base (reference quantity) and aggregate over a subset of the investments. The concentrations are usually calculated using market value weights.

Rating rules

Rating rules check the creditworthiness of an issuer of a security or a counterparty in an active forward transaction or an ongoing transaction (settlement).

Derivative rules

The rules for investment guidelines that relate to derivative instruments are carried out to determine whether the derivatives used are permissible, whether their market risk exposure is within the expected range, whether their coverage is adequate in terms of payment and delivery obligations, whether the issuer exposure is adequate, whether the back-testing confirms the intended investment target and whether the hedging transactions are sufficient.

3.3 Contract Implementation

In the course of the growing regulation of financial markets, fund companies and now also asset managers must be able to show supervisory authorities and clients that they are carrying out a pre and post trade review of internal and regulatory investment guidelines. Specifically, this means systematically translating the above guidelines into a rule-based software tool. The automated prevention of policy violations that result from trading

activities is of crucial importance for all parties involved. The penalties that stem from a single violation as well as the risk of a loss of reputation can be enormous. Unsuitable platforms and software solutions can also result in expenses that endanger business. It is therefore essential to select software tools that are specifically tailored to the contracts of companies and asset managers - which efficiently implement the internal and regulatory investment guidelines in them. The central role is played by a well-defined project plan - from the selection of the right software to the test phase and going live. The following explanations are based on the banking standard.

3.3.1 Impact Assessment of Contract Content

Understanding the exact meaning of each contractual agreement is important for the implementation in the control and the assessment of the liability risk.

Open, imprecise or insufficiently detailed formulations involve process risks and result in complex controls. For this, the portfolio manager has a wider scope when making investment decisions to achieve investment performance. The more stringent the agreements, the more important the pre-trade control for compliance with the investment guidelines becomes. This trade-off must always be considered when the contract is concluded in order to optimize operational costs, product profitability and liability risks. It is advisable to use proven wording when drafting the contract in order to quantify consequential costs and risks. Expertise is often required that a corresponding service provider can offer. Criteria for the selection are long-term experience, rule packages and rule construction in the standard package as well as platform availability. An expert follows the trends and changes in regulations with regard to the control of the investment guidelines and thus reduces considerable additional operational expenditure. This is hardly avoidable due to the liability risks expanded with FINIG, FIDLEG or CISA. Useful contract content and drafting recommendations:

- Designation of the contractual partners
- Limitation of liability and place of jurisdiction
- Indication of the period of validity of the contract
- Order volume
- Reporting obligations
- Charges and administration costs incurred
- Tax aspects
- Risk profile of the investing party
- Specification of the investment review process
- Regulatory obligations
- Investment requirements

3.3.2 Check Completeness of Data

The aim of data validation is to ensure that all the information required for checking the contractual agreements is available: complete, up-to-date and connected in context. The following checks are recommended:

- Position coordination with the custodian bank statements
- NAV calculation in comparison with aggregation of the valued positions
- Control of the instrument identification (security, ISIN, share class, etc.)
- Control of the valuation currency (price, quote) of the instruments
- Control of the term and maturity of the credit instruments
- Control of the counterparty for OTC products
- Issuer of fixed income and money market instruments
- Coordination of transactions with the trade statements and lists of liquidity movements

These validations can of course be carried out on a target platform for reporting and investment guidelines control. This automation happens along with the automation of the investment guidelines check and logically occurs immediately before it. The following techniques are used for validation:

- During position reconciliation (a) the components are counted, (b) the availability of the identification features (symbols, identification numbers, classification attributes) is checked, (c) the transactions made are compared with the items of the previous check period and (d) the balance differences based on the transaction amounts.
- During inventory value validation (a) one checks how current the individual prices of the positions are, (b) checks the number of items based on any corporate actions, (c) checks the market values based on price and quantity, and (d) compares the total market value of all positions with the net asset value.
- In the case of reference data validation (a), the up-to-dateness of the reference data is ascertained, (b) it is checked whether all positions are classified, and (c) it compares aggregated weights and values with values of aggregated reference target values.
- During the market data validation (a) one checks whether valuations - eg. Prices or ratings - exist for each position, (b) compares the differences to the previous test period and (c) explains the deviations using tolerance filters and market events.

3.3.3 Collection, Preparation and Use of Data

When accepting a mandate and changing contracts, data collection is necessary. At the beginning of the periodic review of investment guidelines, reference and market data must be recorded and an inventory or transaction list must be prepared together with all the necessary master data. While the effort for preparing the stocks and transactions is taken into account and mainly based on wage costs, the costs for the provision of market and reference data tend to be higher and consist of labor and data usage costs. The search for necessary reference data can be rationalized by using systems from data providers, of course, for license fees. Alternatively, the best way to cover costs is to get these data services from a provider. Furthermore, the unlicensed use of data may be illegal and caution is advised when using certain data provider systems. The usage agreements show what data may be used for and whether an export to another system or distribution (reporting) is permitted.

3.3.4 Transformation of Contracts into Rules

For the transformation, the contract must be broken down into components - sentences, parts of sentences or expressions - which represent a (closed), investment-relevant and verifiable issue. Incidentally, this also applies to the regulations concerned.

1. Closed issue: This means a clearly defined scope (scope) in which the restriction applies.
2. Investment-relevant facts: Non-mandatory investment goals and conditions, product-irrelevant information or target process descriptions are not hard criteria for the portfolio manager and therefore of little importance with regard to investor protection.
3. Verifiable facts: Numerical specifications and quantity relations such as e.g. are easy to check. allowed asset classes (instrument types), credit ratings or broker choices.

It is also possible to check requirement criteria for external managers and products - typical due diligence analyzes. To do this, compiling a list of contract criteria and defining enumerable answers, which are classified beforehand as sufficient or insufficient is required. These ratings should not change during the term of the contract. It proves sensible to write down the details of the transformation of the contract as this provides proof of how the original implementation took place. A library of rules is often available for the transformation - paying attention to their documentation simplifies the coding and saves a lot of time, since this activity takes place frequently and can only be automated to a limited extent.

Criteria for building a library of test rules are:

- Largely standardized investment guidelines (clear transformation)

- Regulatory or thematic requirements (identical examination)
- Identical accounting processes of the same transaction types (standardized data)
- Similar scope of investment guidelines (simplified, transformation)

Impact Assessment of Contract Content

Every matter from the contract and every relevant requirement from the regulations is now assigned to a rule from the library based on the rule descriptions. If a contractual agreement or regulatory provision does not match at least one existing rule, a new fact-checking rule must be created. In doing so, the set theory, the scope of the instruments, positions or transactions should be formed exactly. The correct calculation formula should be created along with the data attributes - these details are to be described with the general conditions.

The assignment of a restriction rule to a passage has high importance with regard to the duty of care, professionalism of advice and finally the quality in the form of the liability risk. Correct allocation largely assigns the risk to those responsible for the corresponding work step in the value chain. Conversely, an incomplete mapping of the facts from the contracts and regulations may involve further responsibility. The consequences of the breach of due diligence are more serious.

These risks can be minimized by defining the contractual definition precisely, in detail, completely and in a binding manner. However, this can have a significant impact on investment opportunities and thus portfolio management. Here are a few examples: The basic principle is: the more speculative individual investments are, the more precisely the restrictions of these business cases must be worded. After all, such investments involve a higher risk of loss. Once the facts of a contract and the regulations relevant to the mandate have been transformed into test restrictions and classified according to risk, the set of rules can be tested using data.

Collection, processing and use of data

An initial set of master data, reference data and market data are recorded according to the requirements of the test restrictions. The preparation process should be carried out according to the documentation and the data acquisition should be checked for legality.

Check completeness of data

The data collected must be validated for completeness and quality. Particular attention should be paid to the data to be collected manually. If the data is incomplete and consistent, a restriction check is worthless.

Testing and going live

After that, the test rules are to be applied to the data and the results are to be checked using manual parallel calculation. The interim and final results of the tests are documented for acceptance and kept for internal customer documentation.

Check the results

If all results are correct according to the expected values, the test run has been passed and the mandate or product is ready for the routine periodic investment guideline check. Otherwise the rules have to be adjusted and tested again until the result is correct according to expectations.

3.4 Investment Guideline Monitoring

The most visible form of investment controlling for most stakeholders is investment guideline monitoring. In practice, this means the automated pre and post trade monitoring of contractual customer restrictions, regulatory and operational rules as well as internal risk limits that are checked by an investment guideline monitoring system. Despite the growing importance of monitoring investment guidelines in risk management, there is currently a lack of industry-wide standards for the design of operating models, data bases, standard definitions and the handling of interpretative questions. One of the reasons for this is a missing or inadequate interface management of affected systems and stakeholders. Close cooperation with the client is therefore essential - starting with the coding of the investment guidelines using a systematic rule universe, to reporting/correcting violations of the guidelines, and all the way to communication with the relevant strategic and regulatory stakeholders.

Investment Reporting

Customers and stakeholders from financial service providers are increasingly demanding detailed reports on the performance and risk analyzes of their investment portfolios. In addition, regulations such as UCITS IV, MiFID II and now also FINIG and FIDLEG require precise guidelines with regard to risk management and risk reporting. The quality of the reporting is crucial and can be ensured by demonstrating compliance with best practice standards and internal guidelines. Particular attention should be paid to customer reports regarding regulatory and contract-specific guideline violations as well as customer-specific utilization reports.

Chapter 4

Design

This section covers core design decisions concerning the architecture of ARVV. The instantiation of this design architecture is addressed in the following Implementation section. Our ARVV Proof-of-Concept (PoC) is based upon an implementation which is primarily comprised of following 3 distinct parts:

1. Standardization and Refinement of Data
2. Data Collection and Validation
3. Verification Analysis and Translation into Rules

In order to encompass all the above core requirements of ARVV, we disclose our abstract design components below to the reader:

- Data Refiner
- Data Validator
- Data Serializer/Deserializer
- Data Verifier

4.1 Data Refiner

In any ARVV task where reliance on user input is critical, it is important to first clearly organize the required data inputs appropriately into like-groupings. These like-groupings can be based on qualitative similarities or any other method, which allows the user to input their data in a clear and coherent manner. For our application, we organized our data by looking at Investment Schema's such as those discussed in Section 2.4. By doing this, potential logic errors and the collection of redundant data is minimized. Once the appropriate sections are identified, it is also imperative to identify any actions that the

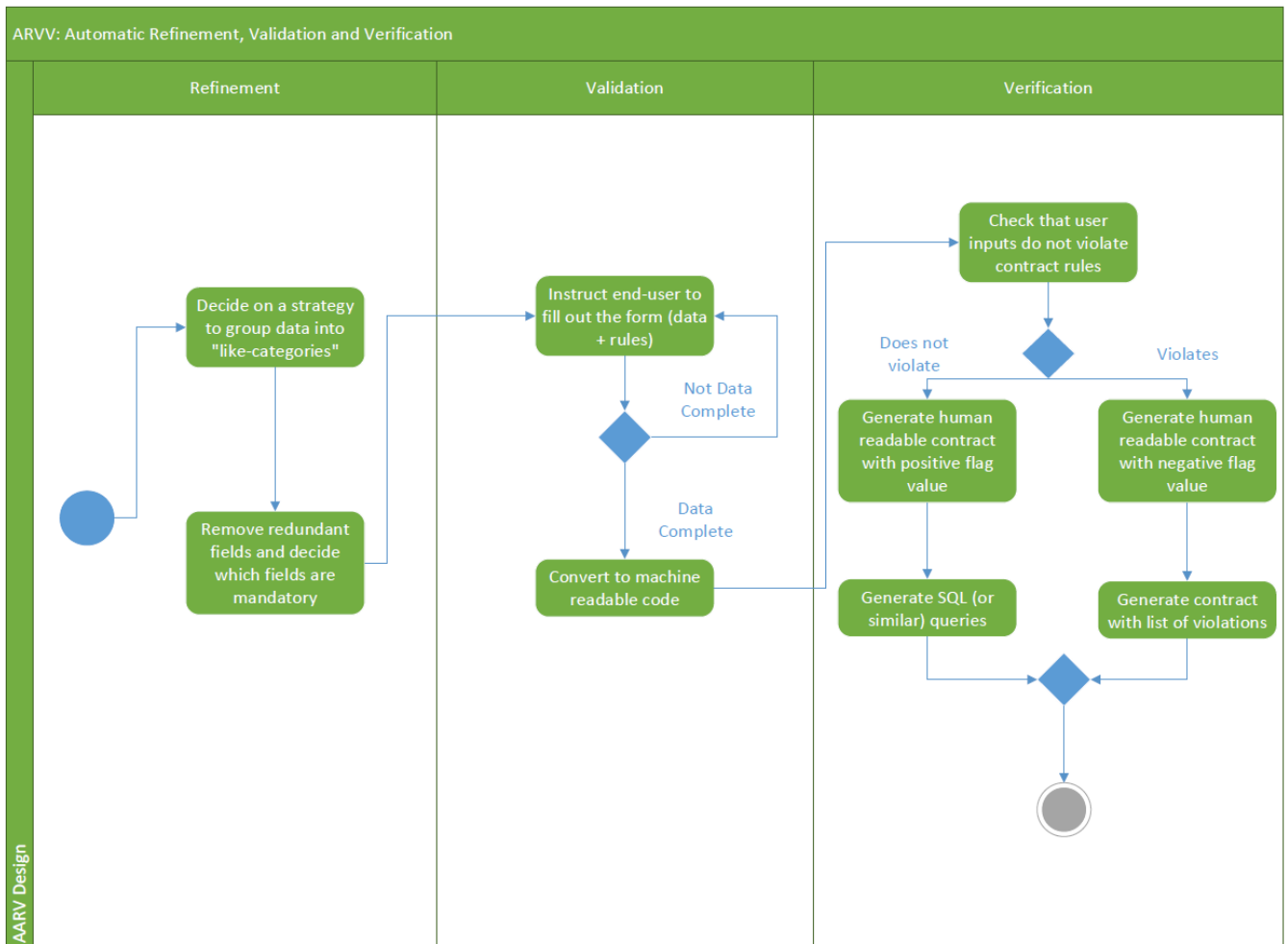


Figure 4.1: ARVV General Design

user may be allowed to take during input; for example, considering whether the fields are always the same type or size or if they are immutable or not is a good starting point for decisions regarding whether or not to include functions such as: *add*, *delete*, *edit*, *move*.

By focusing on the refinement of data - one can ensure that the data is both precise and uniform across the system as it is arranged in such a manner where the end-user is unable to cause logic errors. For example, if a user is from Country A, it does not make sense to provide a list of cities that are outside of Country A. Not only does this refinement process prevent users from entering their inputs erroneously, but it also at the same time benefits an ARVV system in the subsequent steps since less verification checks must be performed (*e.g.* there is no need to verify that the selected City belongs to Country A).

4.2 Data Validator

The aim of the data validator component is to ensure that all the information required for ARVV is filled out by the user as intended. In other words this means data completeness

- complete, up-to-date and connected in context. The data collected can be validated for different aspects and largely depends on the application field where ARVV is applied. Nevertheless, a good starting point is to ensure completeness of required fields with the appropriate data type. If this validation process is not included in the implementation, the verification step does not add value to the system - since no useful response can be returned when mandatory fields are incomplete and/or contain the wrong data type.

4.3 Data Serializer/Deserializer

Sending the contract for verification is a multi-faceted task. Regardless of the size and type of data that comes from user inputs, the end goal is to have these user inputs translated from machine readable code to transition into something that is part of a human-readable contract (or similar) via a verification system. Therefore, depending on the intricacies of the application and how many times something may need to be (re)verified, a form a data deserialization/serialization will need to take place. In this design section and in our PoC, we employ the JSON format as our discussion tool since there is evidence (Section 2.3) that it is superior in terms of performance when it comes to being parsed, being serialized and being deserialized in comparison to other technologies (such as XML).

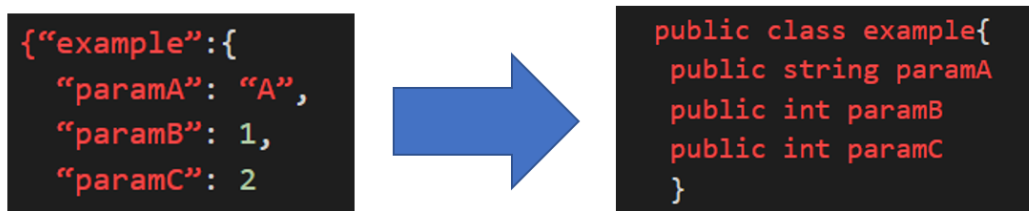


Figure 4.2: JSON Serializing

User inputs are written to a JSON object file and subsequently mapped to our verification system. On an abstract level, this can look something along the lines of Figure 4.2.

However, as the original JSON file is not something that can be easily transmitted to a verification system, a process of serialization must occur. In other words, we need to convert the in-memory data structure into a series of bytes (*e.g.* ASCII). This can be something as simple as using a `JSON.stringify(object)` function in the case of JSON usage in an applicable design. Once a readable data-structure is created, it is possible to assign values to specific variables and functions which do the final verification step. We also do the opposite step (deserialization) when we collect user inputs in the form of strings to store in a JSON object file (in case a contract does not need to be immediately verified, but simply stored in the database).

4.4 Data Verifier

On the last stage, based on a ruleset and after doing some logical operations, ARVV has to conclude whether contract is,

- a) compliant with the rules or,
- b) non-compliant with the rules

We outline our own verification implementation with respect to financial contracts in the subsequent section, along with our usages of Controlled Languages and Robotics Processing Automation technologies, but here we will simply discuss the core design concept of this section (so that the reader can apply it to any ARVV application) - the verification flag. We also include a few of our rulesets as an example to the reader at the end of this section.

4.4.1 Flag Value

We can conclude whether or not verification has successful or not through a Boolean value which we denote as the aforementioned *flag*; this value simply states whether or not the contract is following all the rules needed to be complaint or not (assigning True and False respectively). As we previously concluded with the Validator component that our contract was data complete, at this design stage, ARVV will always generate a human-readable contract - however the flag will be indicative of whether or not a contract is compliant or not and should be visibly present on the contract. The design decision for generating both compliant and non-compliant contracts is because in certain applications, having non-compliant contracts in a human readable format may serve a beneficial purpose (*e.g.* sending to compliance department for a manual exemption request to the ruleset); however, the reader is always able to make modifications accordingly to fit their desired ARVV task.

4.4.2 Examples of Rulesets

Bandwidth rule

Often there are ranges of upper or lower limits for a certain subset of user inputs. These minimum and maximum values can be defined absolutely or relatively - with reference to another (sub) quantity of the inputs or a defined reference quantity.

Bandwidths consist of an upper and a lower limit. If only a single limit is specified, this must be exceeded or fallen short of, depending on the type - otherwise both limits must be respected.

Target rule

In contrast to the bandwidth rule, a target value is targeted and an accepted symmetrical deviation is defined via a tolerance. If the aggregation of a certain subset of user inputs results in an amount that is within the described range, the test result is positive.

Deviation rule

The deviation rule is related to the target value rule. An accepted symmetrical deviation is defined by means of tolerance. If the aggregation of a certain subset of items results in an amount within the range described, the test result is positive.

Chapter 5

Implementation

Our Proof-of-Concept (PoC) is constructed by leveraging Angular CLI v8.3, a back-end database as well as Python 3.8 (as our main automatic verification tool)[1][3]. The goal is to show how Robotic Processing Automation, together with Controlled Language processing is able to substantially improve the daily challenges of stringent documentation requirements for portfolio managers in Switzerland. This work is especially important given the new documentation regulations of Finig and Fidleg that were applied to the industry at the beginning of 2020. More importantly, our PoC backs up the related work we saw in Chapter 2 in which it was clear that a RPA design like this is not only possible, but also superior when it comes to the enforcement of investment guidelines rules - preventing both active (noncompliance with investment restrictions) and passive errors (absence of data completeness) simultaneously.

5.1 Data Standardization and Scrubbing

With a large dataset of over 50,000 line items provided to us privately from the SIX Stock Exchange, we isolated the most important investment attributes that cover all the main Asset Categories currently used by Swiss domiciled financial institutions. By doing this, our PoC covers directly what similar competing (but manual-effort) contracts already have. In other words, we take the most common asset categories that asset managers have to manually verify in today's industry to demonstrate the superiority of ARVV. However, as this is only a PoC, the tool does not encompass the entire universe of all possible investment categories and their attributes/restrictions. We do however, show that it is possible through our data scrubbing efforts (of one the Swiss Stock Exchanges datasets) presented during our midterm presentation.

5.1.1 Data Scrubbing and Refinement

In this section we are going to show how we have implemented the concepts from our framework discussed and defined in Chapter 3. In the aforementioned section, we have

discussed the different types of securities and their interdependence. There were many hidden challenges that came up with the implementation of the 1st design component - data refinement. An example of one such challenge native to our implementation is that securities can either be an instrument or an asset class or both simultaneously. For instance, equities can be an instrument and an asset class while derivatives are only an instrument through which can be bought via different asset classes such as equities. We therefore first defined a way to group separate instruments and their respective combinations together into asset classes. Next, we separated the investment universe into 8 different Asset Categories and followed them up with 7 different Investment Instrument Categories:

Asset Category	Instrument Categories
Equities	Structured Products
Fixed Income	Fixed Income Instruments
Commodities	Money Market Instruments
Real Estate	Currencies
Liquidity	Derivatives
Money Market	Equity Instruments
Alternative	Funds
Other	

Table 5.1: Asset and Instrument Combinations

Recapping the previous example: An equity can be an asset class and an instrument. The same issue is valid for asset classes as well (*e.g.* fixed income, money market, etc). The Other asset classes are commonly acquired via derivatives, funds and structured products. The asset class Other was created intentionally so that a security can still be accompanied by our model provided there is a Title or ISIN - even if we are not aware of other specifications. These specifications are sometimes tied to a future date so its necessary to have a separate asset category while other investments are being monitored. Once the date has been reached, the specifications are available and the security will change from Other to the actual appropriate asset class. For example if a stock pays equity dividends via rights, the rights itself do not have any specifications and will first be put in the asset class Other. However, once the right is exercised and equities are bought, the right disappears from the asset category and will change into the Equities asset category - all of which is accommodated with the above model.

However, a trade off of the above mentioned asset categories and instruments model is that alone it is not granular enough to be able to capture with absolute certainty all restrictions in investment guidelines. We therefore endeavoured to apply more strict identifiers to our model. From the data which we were provided by the SIX, we extracted 187 different descriptions for various securities which we have applied to our defined asset categories and instruments in order for it appear in our PoC. Moreover, the SIX provided us a file with over 55,000 attributes regarding the identification of securities. We trimmed this database to around 1,200 attributes, corresponding to about 10 different tables. These tables include ratings, currencies, indices, listings, countries of domicile,

countries of risk, sectors, stock exchanges and different interest rate calculations. Due to this, we are able target specific identifiers of securities which are necessary to write and verify future database queries.

An example of how we can identify a security - for instance, UBS (UBSN) looks as per the following characteristics:

Asset Category: Fixed Income:

- Instrument: Fixed Income Instrument
- Description: Corporate Bond
- Rating: BBB
- Currencies: CHF
- Indices: SMI
- Listings: Yes
- Countries of domicile: CH
- Countries of risk: World
- Sectors: Banking
- Exchanges: Swiss Stock Exchange
- Convexity: 0.04
- Duration: 5

Lastly, as discussed at the beginning of our paper, we have checked and confirmed these implementations via state of the art contracts currently employed by various domestic financial institutions.

5.1.2 Database

We create a backend REST service to deliver some data in JSON (JavaScript Object Notation) format to the contract verification system. To do this in our PoC, we leverage a lightweight service called: JSON Server [2]. JSON Server is a REST API with CRUD operations and is very lightweight and fast performing for our PoC. It also contains the `-- watch` parameter which ensures that the server is working constantly and looks for file changes and updates in real-time. For us, this means that contracts can constantly be re-verified upon being edited. Furthermore, we are able to make `POST`, `PUT`, `PATCH` or `DELETE` requests where necessary with ease. For the more advanced user, querying the database is also possible in real-time by extending parameters of a URL to the appropriate query. Lastly, we are also able to make use of sorting, searching and pagination, which is helpful for our PoC's transition to a real product.

5.1.3 Form Builder

Our Angular FormBuilder setup ensures that contract validation is seamless. As outlined in our Design goals, ARVV does not allow a financial contract to advance to the verification stage if some fields are missing or invalid - for example if an Asset Manager omits a required field for a specific investment type. Not only does this step reduce errors, but it also removes strain and complexity on our verification tool. This is because it does not send contracts which it knows for certain will be invalid and unable to be verified according to the stipulated regulations. In Angular, we use the FormBuilder class to create a persistent way to capture errors on entry as well as missing values in general. It gives us the following functionality for validation:

- a) highlights fields with errors dynamically
- b) disables the Save button until all the needed data is entered
- c) provides inline prompts while the user is typing in a field

Add Country Restrictions

Country Restrictions 1

Risk Country	CL Chile
Minimum Amount of Above Country	(%) Amount is required
Maximum amount of Above Country (%)	25

Save Load Data

Figure 5.1: No saving without validation

5.1.4 JSON Schema

We make the use of JSON due to its lightweight format for storing and transporting data. It is the notation we use in order to send data from our webpage form - to our contract verification system. It is also easy to interface with many existing in-house financial systems as it is self-describing and easy to serialize and deserialize as we saw in the former sections.

We create our own custom JSON Schema, which is able to describe our data format, provide clear human-and-machine readable documentation and also check data dynamically - which is useful for our validation stage as it ensures the quality of the data the client submits. Where possible, we constrain data types in our PoC (*e.g.* providing certain investment instruments in a dropdown instead of a free-response) in order to demonstrate that this additive step is easy to implement given the requirements of the organization. We also create respective alerts when certain user inputs do not correspond to the model.

```
{
  "InvestmentTitle": "ENJOY",
  "AssetCategory1": "Equities",
  "InvestmentType1": "35 | Private Equities",
  "duration": "None",
  "convexity": "None",
  "minSize": "0",
  "maxSize": "50",
  "sectorExposure": "Multi-Sector Holdings",
  "countryExposure": "CL|Chile",
  "riskLevel": "Intellidex Indices",
  "ratingList": "BB",
  "currencyExposure": "Chilean Peso | CLP"
},
```

Figure 5.2: Excerpt from our JSON Schema

```
for i in range(len(p['json_schema'])):
    investmentTitle = str(p['json_schema'][i]['InvestmentTitle'])
    assetCategory = str(p['json_schema'][i]['AssetCategory1'])
    investmentType = str(p['json_schema'][i]['InvestmentType1'])
    duration = str(p['json_schema'][i]['duration'])
    convexity = str(p['json_schema'][i]['convexity'])
    minSize = str(p['json_schema'][i]['minSize'])
    maxSize = str(p['json_schema'][i]['maxSize'])
    sectorExposure = str(p['json_schema'][i]['sectorExposure'])
    countryExposure = str(p['json_schema'][i]['countryExposure'])
    riskLevel = str(p['json_schema'][i]['riskLevel'])
    ratingList = str(p['json_schema'][i]['ratingList'])
    currencyExposure = str(p['json_schema'][i]['currencyExposure'])
    all_currencies.append((str(p['json_schema'][i]['currencyExposure']),str(p['json_schema'][i]['minSize']),str(p['json_schema'][i]['maxSize'])))
    all_countries.append((str(p['json_schema'][i]['countryExposure']),str(p['json_schema'][i]['minSize']),str(p['json_schema'][i]['maxSize'])))
```

Figure 5.3: Post-Serialization Processing

5.2 Contract Verification and Generation

5.2.1 Controlled Language Processing

Our Controlled Language (CL) processing enables us to automate semantic analysis of the data we gather from the portfolio manager's corresponding inputs. More specifically, we have a *controlled* universe of vocabulary which is relevant to the financial contract in which our subsequent RPA process can understand, group and validate for correctness. Even though we are restricting the universe of available investment products, this does

not impede the Portfolio Manager’s job as new investment products can always be added to the universe upon approval. More importantly, by adding a specific term or instrument to our universe, we do not have to modify existing groupings; instead we simply add to an existing asset category as all the categories which already hold an existing legal framework are already included.

```

6  equities = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Equities] ' \
7          'and related instruments [Investment type related to Equities] listed in [Country of Risk] and der
8  commodities = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Commodities] ' \
9          'and related instruments [Investment type related to Commodities] listed in [Country of Risk] a
10 money_market = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Money Market] ' \
11          'and related instruments [Investment type related to Money Market] listed in [Country of Risk]
12 fixed_income = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Fixed Income] ' \
13          'and related instruments [Investment type related to Fixed Income] listed in [Country of Risk]
14 alternatives = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Alternatives] ' \
15          'and related instruments [Investment type related to Alternatives] listed in [Country of Risk]
16 real_estate = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Real Estate] ' \
17          'and related instruments [Investment type related to Real Estate] listed in [Country of Risk] a
18 liquidity = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Liquidity] ' \
19          'and related instruments [Investment type related to Liquidity] listed in [Country of Risk] and c
20 other = 'at least [AAmin]% and no more than [Aamax]% in [Asset Category: Other Asset Classes] ' \
21          'and related instruments [Investment type related to Other] listed in [Country of Risk] and denominat
22

```

Figure 5.4: An excerpt of predefined CL phrases for each major asset class

5.2.2 Python RPA

Our Python program reduces the human manual effort element in which Investment Controllers and Compliance departments must go through in order to be in adherence with specified regulations. At the moment, the PoC we created follows a set of pre-defined rules that we extrapolated from current manual effort contracts - this ensures that the rules we have are highly relevant to today’s investment controlling landscape. Furthermore, these rules work because of our Design Goals outlined in 4.1, where we standardize the possible combinations without hindering the asset managers toolkit.

Our PoC also generates the corresponding JSON text which services as the baseline for the translation of the Asset Management contract. We also show that generating SQL queries for positively flagged investment contracts is feasible by converting JSON into SQL. However, since the tools used today (Simcorp, MIG21, Allocare, Sentinel, Aladdin, Charles River and Bloomberg Compliance) all have their own proprietary databases, generating fully compatible SQL queries is not feasible at the PoC stage. However, with collaboration with an interested firm, this can be tailored into the security master files of their compliance systems by adopting their nomenclature.

5.2.3 Contract Manipulation

We primarily made use of the *python-docx* library [4] (a Python library for creating and updating document files) to generate the investments types set out by the end user in accordance with the validated contract. This allows data sent over the server to be seamlessly integrated into the contract. In the case of a positive flag (discussed in the Design section), we not only generate a contract with the *python-docx* library, but also generate the corresponding *SQL queries* - or in other words, a rule-set which encompasses the asset allocation which adheres to the investment guidelines. With this implementation, an expansion into further work leaves room for machine-learning capabilities as well as adoption of compliance system specific nomenclature.

```

for i in range(len(country_res)):
    for j in range(len(all_countries)):
        if ((str(p['countryRes'])[i]['countryRestriction']) in all_countries[j][0]):
            if (float(all_countries[j][1]) < float(p['countryRes'][i]['minCounRes'])):
                document.add_paragraph("Not enough " + all_countries[j][0] + " exposure. You have " + str(
                    all_countries[j][1]) + "% listed")
                conflicts = True
            if (float(all_countries[j][2]) > float(p['countryRes'][i]['maxCounRes'])):
                document.add_paragraph("Exceeding " + all_countries[j][0] + " exposure. You have " + str(
                    all_countries[j][2]) + "% listed")
                conflicts = True
        break

```

Figure 5.5: document.addParagraph() method

Chapter 6

Evaluation

In this section, we focus on showing why our ARVV model is superior in comparison to the manual effort contracts which are used in today's investment compliance and controlling industry. We present the main benefits along with a usability study and a live use-case example. At the end of the chapter we also address some challenges that relate to our ARVV PoC and possible future improvements.

6.1 Effectiveness, Efficiency and Accuracy

Through our PoC, we show 3 main benefits by validating and verifying a financial using RPA and CL:

1. Less Prone to Errors
2. High Uniformity and Precision
3. Time and Cost Savings

6.1.1 Error Reduction

There are two main types of errors that are made in financial contracts - Active Errors as well as Passive Errors. Our PoC prevents both errors as active errors - noncompliance with investment restrictions forbid the contract from being successfully verified; passive errors - absence of data completeness, prevent the contract from ever being sent to the verification stage and subsequently acquiring the SQL queries for generating a formal ruleset.

6.1.2 High Uniformity and Precision

Our PoC defines a stringent set of rules for the validation aspect of the financial contract as well as the latter verification part. From a design standpoint, this means that even though every contract may have different assets and instruments, the parameters and corresponding related attributes are already known. Not only does this alleviate error, but it also allows Investment Controllers and other interested parties to run additional queries on all the contracts without having to worry about considering the different attributes and dependencies that each contract may have. Furthermore, any model changes that occur are able to be changed in real time via our PoC tool. Our PoC is able to handle changes and editions with minimal editing to the codebase and also without breaking previous stored contracts.

6.1.3 Time and Cost Savings

By translating asset management contracts along with capturing the corresponding investment guidelines into text-preserves, embedding them in portfolio management and compliance systems saves a lot of manual effort work for compliance offers. Investment controlling software commonly use a text-preserve/SQL based system - for example, Simcorp, MIG21, Allocare, Sentinel, Aladdin, Charles River and BloombergCompliance are all based exclusively on the aforementioned systems. However, due to a lack of automatic validation and verification, many contracts are not standardized and each of them is currently manually validated and verified by a firms employees. In other words, each portfolio must be individually coded into the investment controlling software. Our PoC only requires these standardized guidelines to be coded in once and not for each individual portfolio as it is the case today. This results in a large savings which can also be passed down to the client and increase a firms competitiveness in the market. As almost all the investment controlling products available on the market are based on a text-preserve/SQL basis, our idea in the PoC can extend to the industry as it is fairly straightforward to embed into any common Investment Controlling Software System (ISSS).

6.2 Usability Study

The official ISO 9241-11 definition of usability is: *the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use* [19]. Even though throughout our Master Project, we have been constantly in touch with industry professionals who are interpreted in our tool, we wanted to formally quantify the superiority of our tool with a usability study. In other words, we wanted to show just how easy, effective and satisfied Investment professionals were with the tool.

Usability Study Results			
	Manual-Effort Contract	Our ARVV PoC	Comments
Ease of Use	benchmark	higher/better	Appendix B
Ease of Verification	benchmark	higher/better	Appendix B
Validation Messages	benchmark	higher/better	Appendix B
Functions/Capabilities	benchmark	lower/worse	Appendix B
Additional Features	benchmark	higher/better	Appendix B

Table 6.1: Results of Usability Study

6.2.1 Background of Usability Study

In order to confirm that our tool captures the information required and does indeed have added benefits, we conducted a usability study based on the methods discussed in Section 2.5 - namely PSSUQ. We recruited 7 Investment Controlling professionals and gave them a questionnaire which can be found in the Appendix. To rank ARVV, we presented the participants with our tool in conjunction with an industry leading paper based contract. The recruited participants are all people who used the ARVV tool were all versed in the investment controlling/compliance industry. The most salient findings are summarized in the table above - while the entire study can be found in Appendix B.

Discussion of Results

The main goal was to compare the critical components of our tool in comparison to a manual effort paper-based contract. For us, this was Ease of Use, Ease of Verification, Validation Messages, Functions/Capabilities. We compiled the results from the questions and for many, ARVV was a clear winner.

Apart from the critical components, we concluded that the participants appreciate the other features we implemented that are not necessarily present in a paper based contract (prompts, help messages, automated verification). The only aspect in which the PoC lacks is in terms of the system having all the functions and capabilities. This is actually an expected result as this is a PoC and it does not encompass the entire universe of instruments. The universe of instruments is usually unique to each firm and must be tailored to fit the requirements on a case by case basis.

6.2.2 Further Features

In our ARVV tool, our form is not only able to generate a contract, but the database it is linked to constantly allows the investment professional to also access, modify and delete other existing contracts on the go. This all occurs in a centralized way - where upon any form of modification to a contract, it is re-validated and subsequently re-verified according to the then present investment guidelines for that particular contract.

6.3 Use Case

We present a use case where a potential investment professional is creating an Investment Contract. The professional knows their client, their contact information and their overall risk tolerance. The investment professional is tasked with creating an investment strategy that suits the client and is also in compliance with the risk tolerance of that particular client. However, before being able to implement the investment strategy, the contract must be verified for the aforementioned compliance.

There are many pitfalls that the investment professional can fall into. Firstly, they need to make sure that they are not exceeding any of the Asset Category restrictions (*e.g.* no more than $x\%$ in Equities). Then they also need to make sure that they do not violate certain further investment restrictions - for example, the client wishes his portfolio to be invested in at least 10% in European equities, but no more than 25%. Furthermore, the investor may want to diversify their currency risk a bit. Instead of having their portfolio's in only Euro's, the investor may want to have some exposure to the US-Dollar (let's say a range of $[10 - 20\%]$). Lastly, let's assume that the client wishes to also invest a small part of his portfolio into a risky Emerging Market Asset - lets say the requirements are a bond issued in any emerging market, other than Turkey, that is denominated in US-Dollars.

Currently our PoC is able to automatically verify these basic and advanced investment restrictions. For instance, according to the guidelines, verifying compliance for minimums and maximums in certain asset category is not difficult. Making sure that investments are not in certain currencies or countries is also rather straightforward. However, we take it a step further and look into the details of each investment being made and by doing so, we can make complex decisions about whether a certain contract follows the guidelines or not. From our previous paragraph's example, we can imagine that if the investment manager purchased an Index tracking the SP500 (in USD) for 20% of the clients portfolio and also the risky Emerging Market Asset for 3% exposure (also in USD), they would be non-compliant of the investment guidelines which call for $[10 - 20\%]$ of USD exposure. One can imagine how checking for all possible restrictions manually is difficult and cumbersome in a contract that is comprised of many investments.

Sample Guidelines			
	Minimum (%)	Maximum (%)	
Asset Category			
Equities	0	50	
Fixed Income	0	50	
Currency Exposure			
Euro (EUR)	0	50	
US Dollar (USD)	10	20	
British Pound (GBP)	0	50	
Country of Risk	-	0	
Turkey			
Argentina	3	3	

Investments Made			
	Exposure Currency	Country	Asset Category
Stoxx 600 EURO	50 EUR	Germany	Equity
FTSE 100	GBP	U.K.	Equity
S&P 500	20 USD	U.S.A	Equity
Argentina Bond	3 USD	Argentina	Fixed Income

Maximum 20% of USD allowed, but invested in 23% even though investing outside of the USA			
Too much USD exposure, need to decrease S&P by at least 3%			

Figure 6.1: Too Much USD Exposure

6.4 Challenges and Future Work

As our work is largely a PoC, fully ensuring compliance with all the CISA encompassing regulations was not feasible. Instead, we showcased the power of RPA and CL and the drastic improvements it can make on the investment controlling industry by tackling the most pivotal attributes outlined at the beginning of this section. With that said, closer collaboration with interested firms is needed. This will ensure that a tool is developed that encompasses all the rules and regulations under CISA with that particular firms corresponding investment offers.

Furthermore, the adoption of new technology in the finance world in a timely manner may be lengthy and is oftentimes difficult. This is even more-so true when it comes to software systems which directly impact the heart of a firms compliance department. If our PoC is eventually translated into a product for these firms, a lengthy design process will also be compounded by legal scrutiny. This can overall translate into large delays prior to the release of the tool. However, given the success of our PoC and the level of interest which we have received from some parties in the industry, we are confident that the short to medium term development efforts/challenges will be greatly outweighed by the potential long-term benefits.

Chapter 7

Conclusion

Asset management contracts and investment guidelines are both costly (4000 - 5000CHF per contract) and a lengthy development effort to complete them and ensure compliance. Furthermore, there is no one standard for creating investment guidelines today and often-times within an investment firm, many different formats are employed. Asset managers constantly face challenges in the validation (data completeness) and verification (ensuring compliance) of these contracts. By developing a standardized contract based on a pre-defined questionnaire, asset managers are able to save both time and money for their firms.

Our PoC demonstrates that it is possible to standardize such an asset management contract and also in the same process check whether the contract is compliant with the corresponding investment guidelines. We accomplish this by leveraging Robotics Processing Automatic (RPA) and Controlled Languages (CL) in order to validate the contract and also generate the appropriate text preserves in the verification stage. By doing so, we can ensure that asset managers are not overburdening their firm through repetitive compliance processes and also that financial market overseers can concentrate on only one format during the year-end review.

We show that our tool which automatically refines, validates and verifies these financial contracts has the ability to increase the efficiency of financial market operations. This in turn can directly reduce costs for not only the asset manager, but also for the entire firm. Furthermore, the potential for errors and thus the overall risk are minimized - which also translates into savings that can be passed down to the client as the firms legal and operational risk costs are brought down.

Abbreviations

ARVV	Automated Refinement, Verification, and Validation
CISA	Collective Investment Schemes Act
CSG	Communication Systems Group
CNL	Controlled Natural Language
CL	Controlled Language
FINMA	Swiss Financial Market Supervisory Authority
FX	Foreign Exchange
ISSS	Investment Controlling Software System
JSON	JavaScript Object Notation
MDS	Market Data System
MiFID	Markets in Financial Instruments Directive
NAV	Net Asset Value
NLP	Natural Language Processing
OTC	Over the Counter
PM	Portfolio Manager
POC	Proof of Concept
PSSUQ	Post Study System User Questionnaire
RPA	Robotics Processing Automation
SNB	Swiss National Bank
UCITS	Undertakings for the Collective Investment in Transferable Securities

List of Figures

2.1	SBVR Business Language	6
2.2	Employment Impact of Document review	8
2.3	Usability Questionnaire	13
4.1	ARVV General Design	30
4.2	JSON Serializing	31
5.1	No saving without validation	38
5.2	Excerpt from our JSON Schema	39
5.3	Post-Serialization Processing	39
5.4	An excerpt of predefined CL phrases for each major asset class	40
5.5	document.addParagraph() method	41
6.1	Too Much USD Exposure	46

List of Tables

5.1	Asset and Instrument Combinations	36
6.1	Results of Usability Study	45

Appendix A

Contents of the CD and Installation Guidelines

A.1 Angular Form

The first part of the ARVV tool for financial contracts is the form where the end-user is able to input the proposed investments as well as the restrictions that go along with it. The form is setup as an Angular Project. To run it, creating a brand new Angular project using Angular CLI is required. Firstly, install Node and NPM using the link below: <https://nodejs.org/en/download/>. It is recommended to use at least Node version 8.x or greater as well as NPM version 5.x or newer. Once installed, please use execute NODE and NPM in your shell to confirm the versions.

Next, install Angular CLI with the command `npm install -g @angular/cli` and then update it using `npm install -g @angular/cli@latest`. Installing JSON-Server as a REST-API is also needed. To do so, execute the command `npm install -g json-server`. Next, navigate to the AngularProject directory in your shell and execute the command `ng serve -open`. This will start the form. Also, start the REST-API by executing `json-server -watch db.json` from the same path in the shell.

A.2 Python Verification

The verification tool runs on Python. Please install any version of Python 3.x. Next, use pip to install `python-docx` by running the command: `pip install python-docx`. Now, in the PythonVerifier folder, open the program in an editor of your choice and edit the path of the JSON file generated in the previous step for your machine (line 40 in `json_verifier.py`). Alternatively, we included some sample json files that you are already present in the folder; however, depending on project setup, editions to the path may have to be made. Execute the Python program (`json_verifier.py`) and a document called `demo.docx` will appear in the project directory. This will be the results of the ARVV given the JSON files created by the form.

A.3 Contents of the CD

The CD contains the following file tree:

```

/
├── thesis.pdf.....The Thesis in pdf format
├── midterm.pdf.....The midterm presentation slides in pdf format
├── IRC_VDF_01_External.xlsx.....Complete Attribute List
├── Asset.Kategorien.xlsx.....Scrubbed Data for PoC
├── AngularForm.....Source Code Files
│   ├── /src/app/employee
│   │   ├── create-employee.component.ts.....Main Routing Elements
│   │   ├── create-employee.component.html.....HTML of form
│   │   ├── create-employee.component.js.....JS of form
│   │   ├── IEmployee.ts.....Interface for Portfolio Manager (PM)
│   │   ├── IRestriction.ts.....Sets out general restrictions
│   │   ├── ISkill.ts.....Different attributes to include by PM
│   │   ├── IEmployee.ts.....Interface for all restrictions
│   │   ├── ICountryRestriction.ts.....Country Restriction Attributes
│   │   └── IFXCRestriction.ts.....Currency Restriction Attributes
│   ├── app-routing.module.ts.....Angular Routing Schema
│   ├── app.component.html.....Link and Create on Form
│   ├── app.module.ts.....Relevant Module Imports
│   └── db.json.....JSON results generated from Angular Form
├── PythonVerifier.....The Verification Step
│   ├── json_verifier.py.....Verifies the generated json file
│   ├── db1_sample.json.....Sample files
│   ├── db2_sample.json.....Sample files
│   └── demo.docx.....Generated Contract

```

Some files are excluded deliberately from the above file tree as they are mostly configuration files that should not be of particular interest to the reader. They are however, present in the CD submission.

Appendix B

Usability Study Results - PSSUQ

PSSUQ

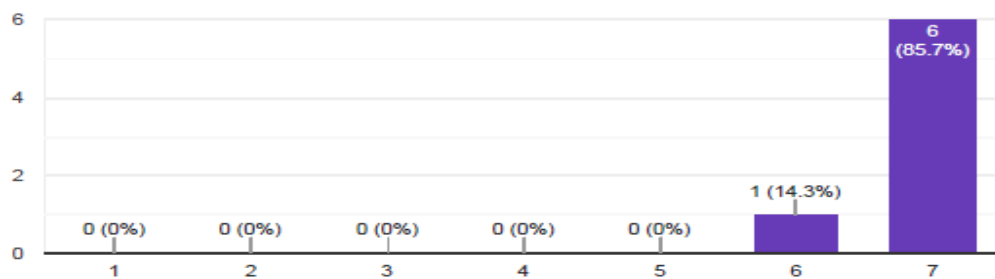
7 responses

[Publish analytics](#)

Overall, I am satisfied with how easy it is to use this system.

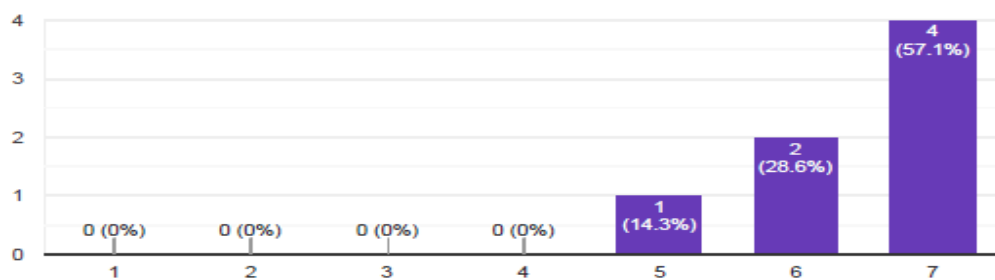


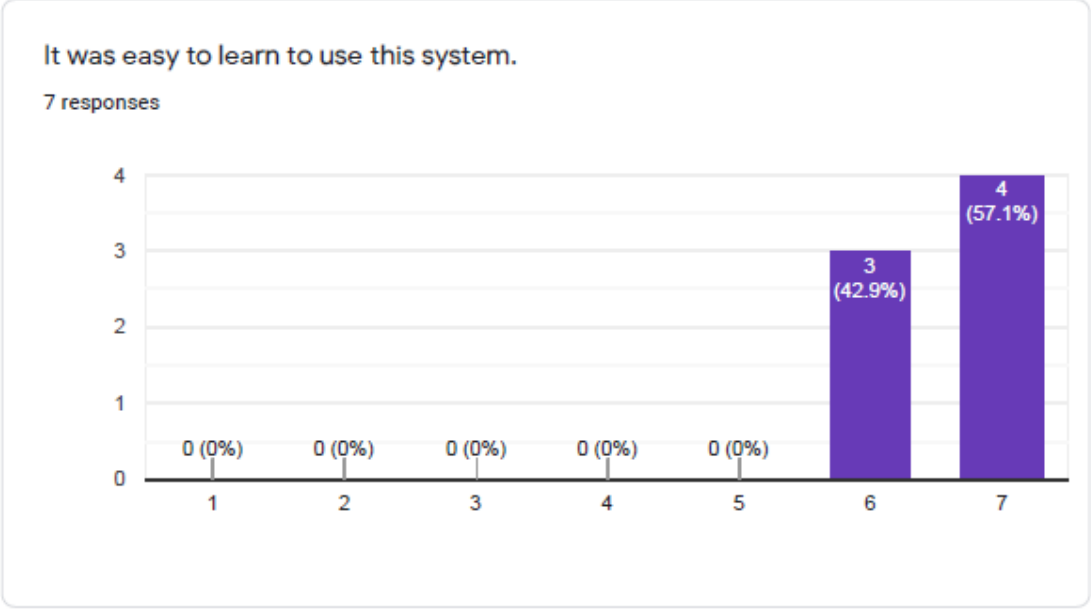
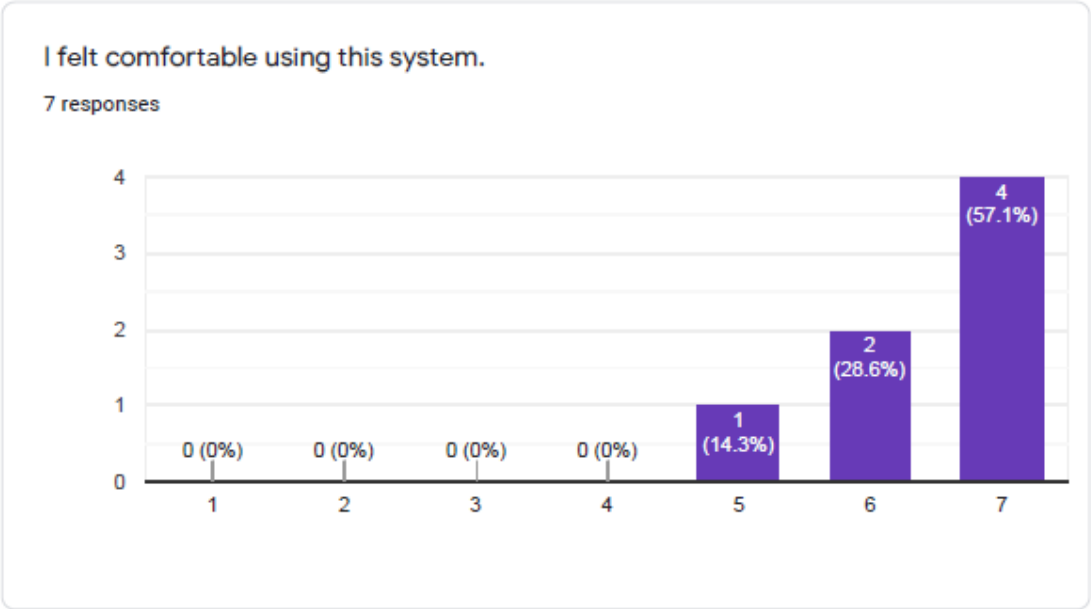
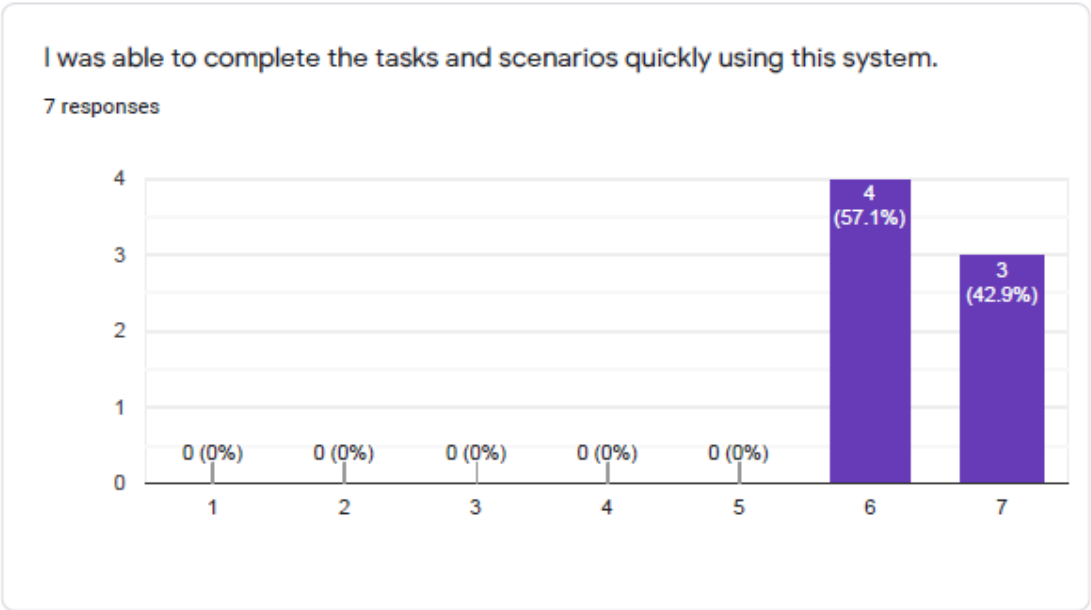
7 responses



It was simple to use this system.

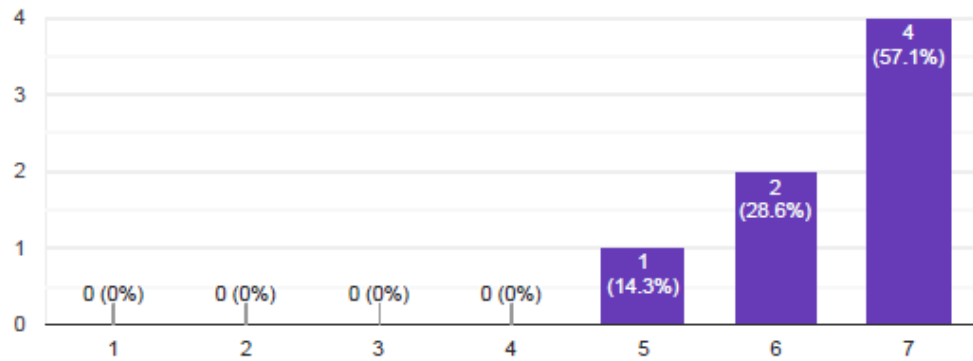
7 responses





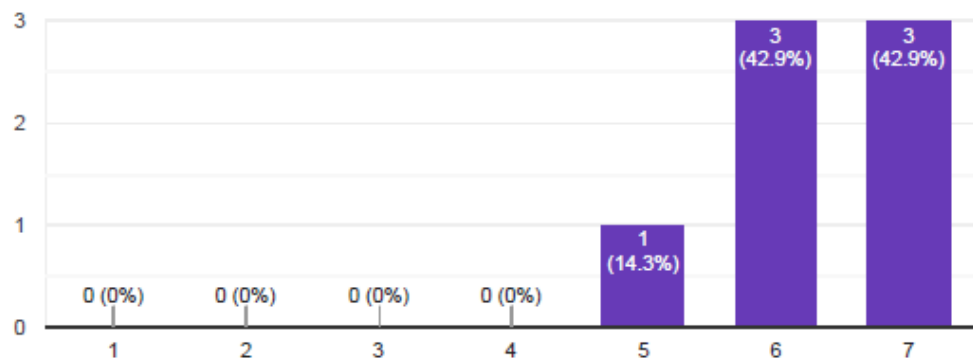
I believe I could become productive quickly using this system.

7 responses



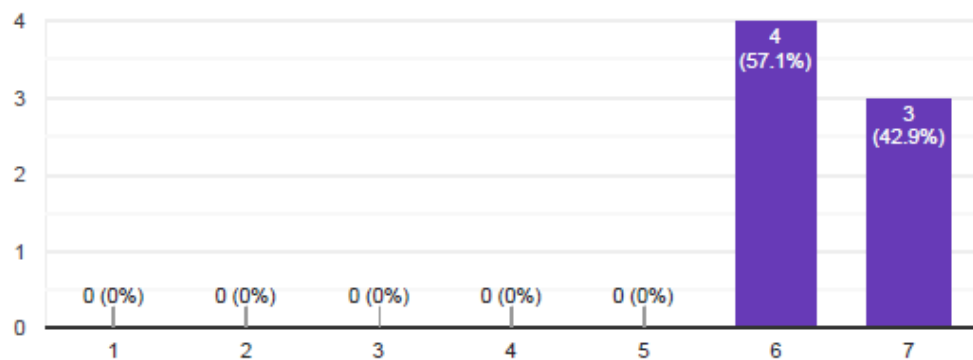
The system gave error messages that clearly told me how to fix problems.

7 responses



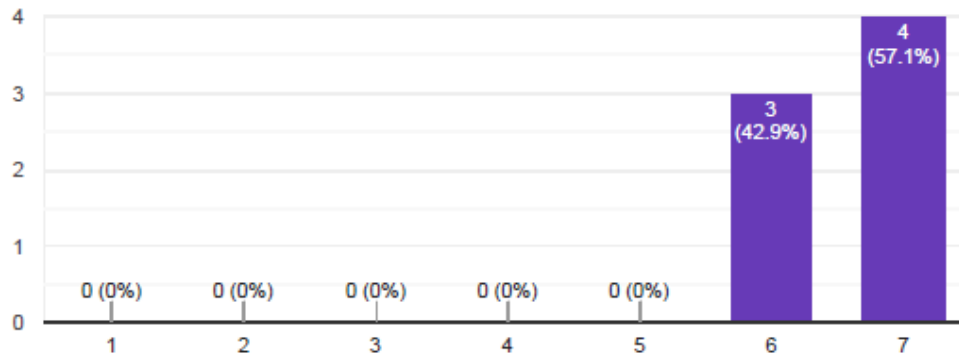
Whenever I made a mistake using the system, I could recover easily and quickly.

7 responses



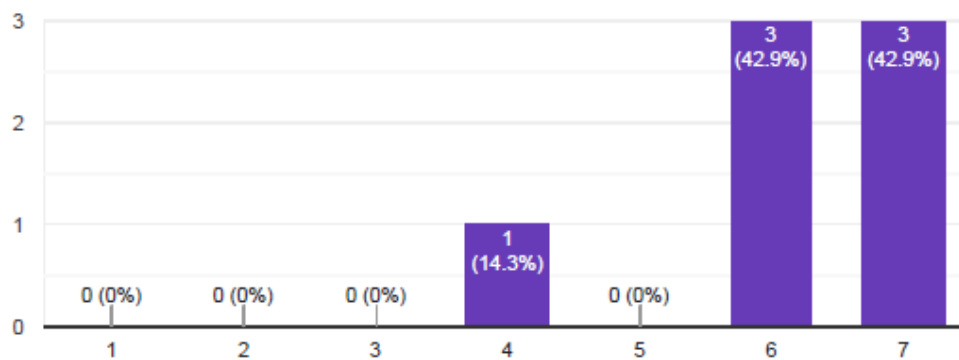
The information (such as online help, on-screen messages, and other documentation) provided with this system was clear.

7 responses



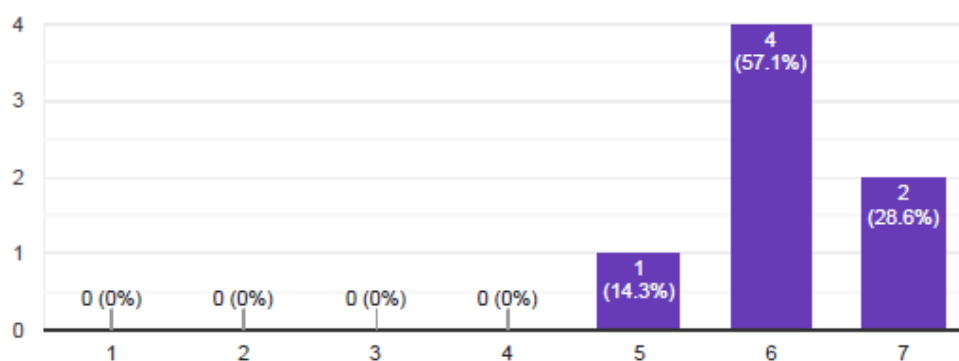
It was easy to find the information I needed.

7 responses



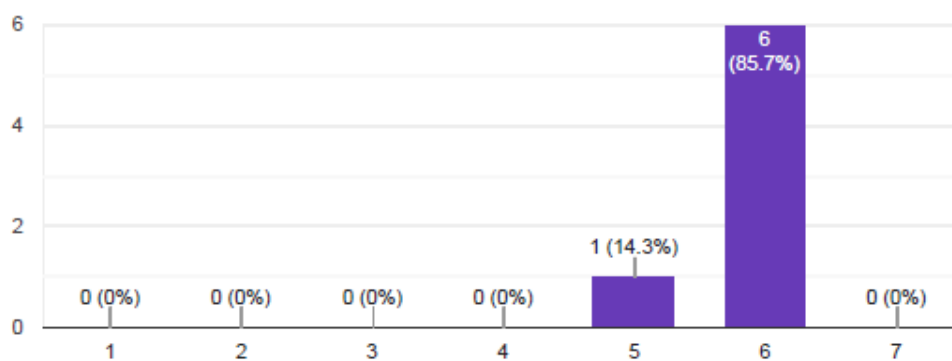
The information was effective in helping me complete the tasks and scenarios.

7 responses



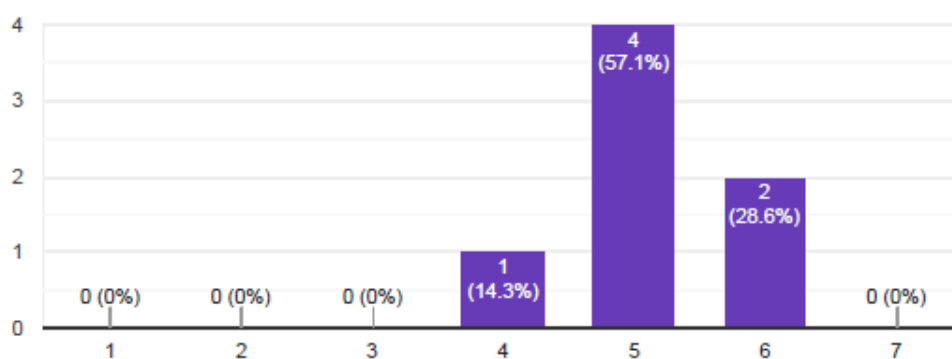
The organization of information on the system screens was clear.

7 responses



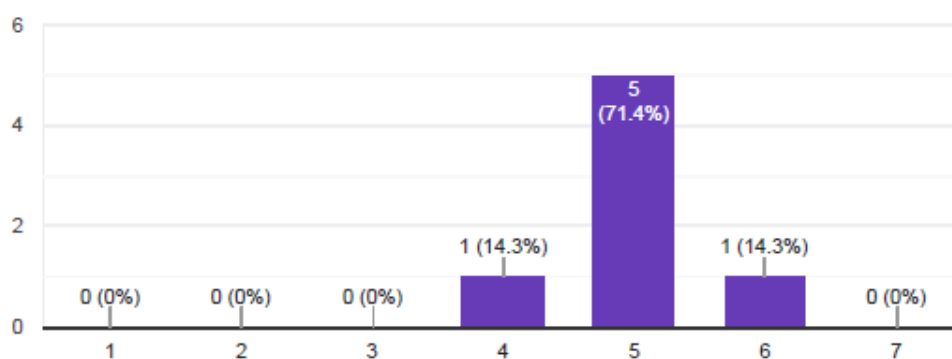
The interface of this system was pleasant.

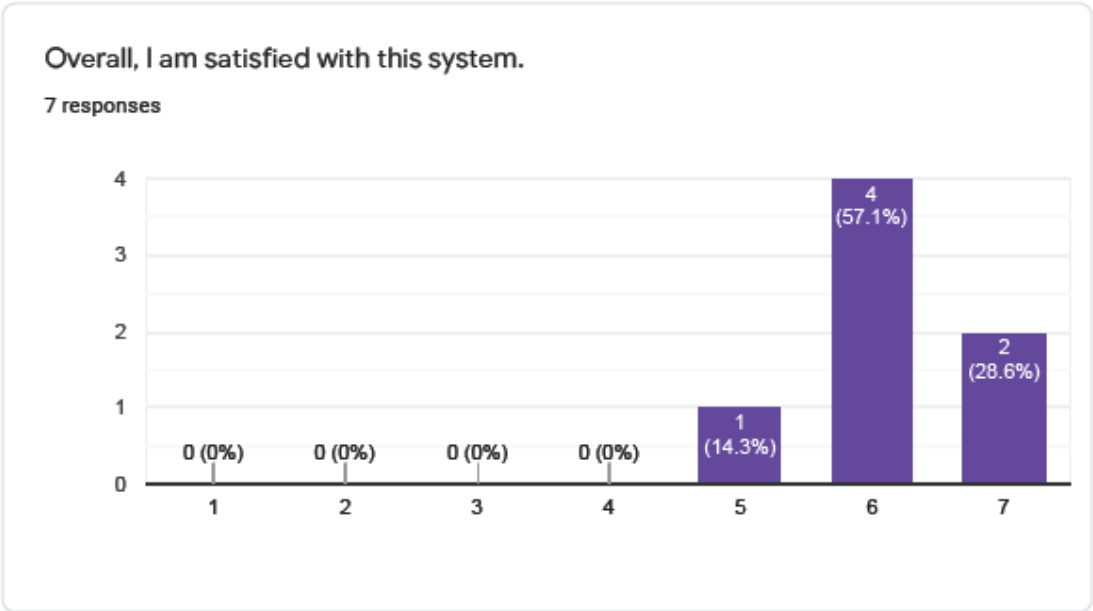
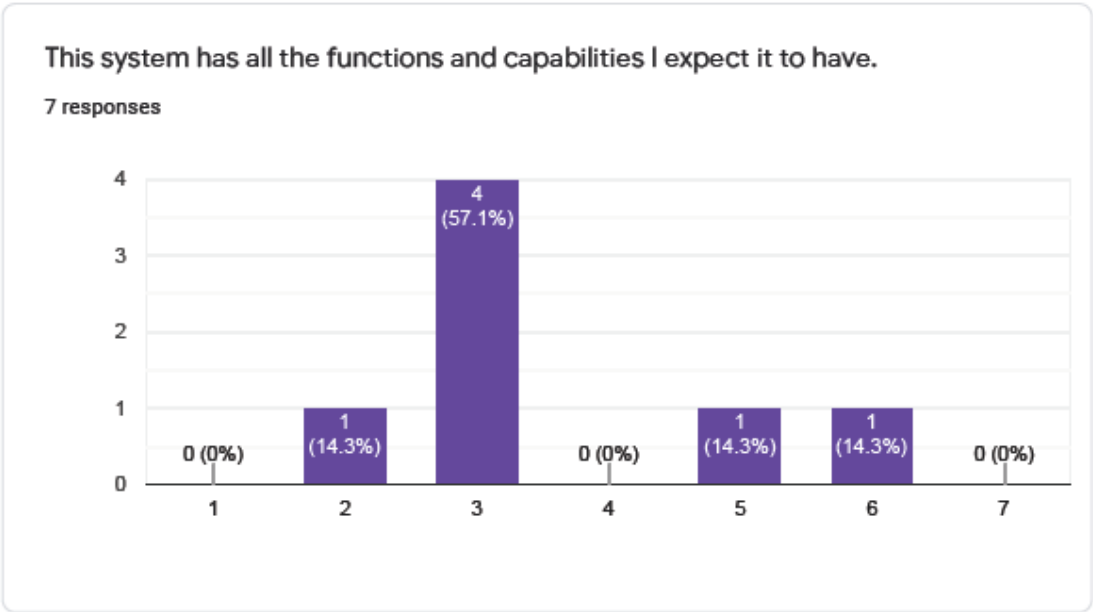
7 responses



I liked using the interface of this system.

7 responses





Bibliography

- [1] Angular8.3, cli <https://www.npmjs.com/package/@angular/cli/v/8.3.6>.
- [2] Json server, <https://github.com/typicode/json-server>.
- [3] Python 3.8, <https://docs.python.org/3/whatsnew/3.8.html>.
- [4] Python docx, <https://python-docx.readthedocs.io/>.
- [5] Ciso finma - collective investment schemes act. 2020 (accessed June, 2020).
- [6] Classification of financial instruments (cfi) - iso 10962. 2020 (accessed June, 2020).
- [7] Finsa and finia. financial services act and financial institutions act. federal department offinanc. 2020 (accessed June, 2020).
- [8] Mifid ii. markets in financial instruments directive (2004/39/ec). european securities and markets authority. 2020 (accessed June, 2020).
- [9] System of national accounts 2008. 2020 (accessed June, 2020).
- [10] Wil M. P. Aalst, Martin Bichler, and Armin Heinzl. Robotic process automation. *Business Information Systems Engineering*, 60(4):269–272, 2018.
- [11] M. Antonelli, D. Bernardo, H. Hagraš, and F. Marcelloni. Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification. *IEEE Transactions on Fuzzy Systems*, 25(2):249–264, 2017.
- [12] Ankit Bharchan and Devesh Bharathan. Relationaljson, an enriched method to store and query json records. *International Journal of Computer Applications*, 98:3–6, 07 2014.
- [13] Steven Davy, Brendan Jennings, and John Strassner. The policy continuum - policy authoring and conflict analysis. in *Special Issues of Elsevier Computer Communications on Self-Organisation and Self-Management in Communications*, 31, 08 2008.
- [14] Andrea De Mauro, Marco Greco, and Michele Grimaldi. A formal definition of big data based on its essential features. *Library Review*, 65:122–135, 03 2016.
- [15] Miki Enoki, J. Siméon, H. Horii, and Martin Hirzel. Event processing over a distributed json store: Design and performance. volume 8787, pages 395–404, 10 2014.

- [16] Han Ping Fung. Criteria, use cases and effects of information technology process automation (itpa). *Advances in Robotics and Automation*, 3:1–10, 07 2014.
- [17] Chris Holder, Vikram Khurana, Faye Harrison, and Louisa Jacobs. Robotics and law: Key legal and regulatory implications of the robotics age (part i of ii). *Computer Law Security Review*, 32, 03 2016.
- [18] Q. Nguyen Hung, Michael Hackett, and K. Whitlock Brent. *Happy About Global Software Test Automation: A Discussion of Software Testing for Executives*. Happy About, 2006.
- [19] Timo Jokela, Netta Iivari, Vesa Tornberg, and Polar Electro. Using the iso 9241-11 definition of usability in requirements determination: case studies. In *Proceedings of HCI2004: Design for life, the 18th British HCI group annual conference, Leeds Metropolitan University, UK*. Eds. A. Dearden & L. Watts, volume 2, pages 155–156, 2004.
- [20] Tobias Kuhn. A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1):121–170, mar 2014.
- [21] Zhen Hua Liu, Beda Hammerschmidt, and Doug McMahon. Json data management: Supporting schema-less development in rdbms. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, page 1247â1258. Association for Computing Machinery, 2014.
- [22] Glenford J Myers, Corey Sandler, and Tom Badgett. *The art of software testing*, pages 144–145. John Wiley & Sons, 2011.
- [23] Phuc C. Nguyen, Thomas Gilray, Sam Tobin-Hochstadt, and David Van Horn. Soft contract verification for higher-order stateful programs. *CoRR*, abs/1711.03620, 2017.
- [24] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT â93 and CHI â93 Conference on Human Factors in Computing Systems*, page 206â213. Association for Computing Machinery, 1993.
- [25] Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, and Clemente Izurieta. Comparison of json and xml data interchange formats: A case study. In *CAINE*, 2009.
- [26] Dana A. Remus and Frank Levy. Can robots be lawyers? computers, lawyers, and the practice of law. volume 3, pages 501 – 558, 2016.
- [27] Tanina Rostain. Robots versus lawyers: A user-centered approach. *Georgetown Journal of Legal Ethics*, 30:559, 2017.
- [28] Jianping Li Yingjiong Zhong Rui Liu Shuang Yang, Kun Guo and Zili Feng. Framework formation of financial data classification standard in the era of the big data. *Procedia Computer Science*, 30, 2014.
- [29] James R. Slaby and Phil Fersht. Robotic process automation: Overview and opportunities. *HfS Research, Lt*, 2012.

- [30] Ernst-Ludwig von Thadden. Long-term contracts, short-term investment and monitoring. *Review of Economic Studies*, 62(4):557–575, 1995.
- [31] John Whiteside, John L. Bennett, and Karen Holtzblatt. Usability engineering: Our experience and evolution. pages 791–817, 1988.
- [32] Adam Wyner, Krasimir Angelov, Guntis Barzdins, Danica Damljanovic, Brian Davis, Norbert Fuchs, Stefan Hoefler, Ken Jones, Kaarel Kaljurand, Tobias Kuhn, Martin Luts, Jonathan Pool, Michael Rosner, Rolf Schwitter, and John Sowa. On controlled natural languages: Properties and prospects. pages 281–289, 06 2010.
- [33] Kamir Yusof and Mustafa Man. Efficiency of json for data retrieval in big data. *Indonesian Journal of Electrical Engineering and Computer Science*, 7:250–262, 07 2017.