

Universidade do Minho
Escola de Engenharia

**Maturing Information management
in compliance with international standards**

Knight Victor

**Maturing Information management
in compliance with international standards**



European Master in
Building Information Modelling

Knight Victor



UMinho | 2022

The European Master in Building Information Modelling is a joint initiative of:



Universidade do Minho

Univerza v Ljubljani



POLITECNICO
MILANO 1863



Co-funded by the
Erasmus+ Programme
of the European Union

November 2022



Universidade do Minho

Escola de Engenharia

Knight Victor

Maturing Information management in compliance with international standards



European Master in
Building Information Modelling

Master Dissertation

European Master in Building Information Modelling

Work conducted under supervision of:

Jose Granja

Marta Campos (tutor in company)



Co-funded by the
Erasmus+ Programme
of the European Union

November, 2022

AUTHORSHIP RIGHTS AND CONDITIONS OF USE OF THE WORK BY THIRD PARTIES

This is an academic work that can be used by third parties, as long as internationally accepted rules and good practices are respected, particularly in what concerns to author rights and related matters.

Therefore, the present work may be used according to the terms of the license shown below.

If the user needs permission to make use of this work in conditions that are not part of the licensing mentioned below, he/she should contact the author through the RepositóriUM platform of the University of Minho.

License granted to the users of this work



Attribution
CC BY

<https://creativecommons.org/licenses/by/4.0/>

ACKNOWLEDGEMENTS

Grateful I am to God Almighty for his grace and strength to embark on this journey. To my beloved family Dr. & Mrs E.D. Knights thank you for all your support and prayers. Romeo, Lemuel, Peter, and Paul a big thank you for all the encouraging calls it meant a lot to me during this work.

A very Big thank you to Marta Campos, I must say your passion and enthusiasm is infectious in a good way. Giving you time and resources to this course has not only supported me through the dissertation but, has shown me that critical thinking and perseverance open doors to new frontiers, which could be just that leap one needs to get to the next level. Power to you and the good work you are doing. I will always remember it.

I would like to express my gratitude to BIM A+ consortium, for supporting me with a fine token which subsidised my expenses.

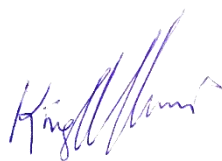
In a special way I would like to thank Oghenevwede Gift Agboro for being the cherry on this journey. God bless you.

Jose Granja, your push, and guidance was helpful through the entire process I am grateful.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

A handwritten signature in blue ink, appearing to read 'Knight Victor', is positioned above the printed name.

Knight Victor

RESUMO

A Gestão da Informação utilizando a Modelação da Informação sobre Edifícios (BIM) é um conjunto de conceitos e princípios recomendados para apoiar a gestão e produção de informação durante o ciclo de vida de um bem construído (EN ISO 19650-1, 2018). A gestão da informação é importante, especialmente em tempos recentes, com um tremendo impulso para a transformação digital. A noção de que a gestão da informação é melhor feita em sistemas robustos de gestão da informação (IMS) é correcta. É justo dizer que a dinâmica global vai no sentido de soluções baseadas na nuvem ou electrónicas mas, por outro lado, muitas empresas podem ser excluídas devido à dinâmica de mercado de acessibilidade em relação ao investimento de capital acessível à sua disposição. Certamente, chegará o momento em que a mudança antecipada para estes sistemas acontecerá, mas a grande questão é o que acontece entretanto? A informação criada, armazenada, utilizada e gerida deve ser organizada de forma bem estruturada, com as recomendações correctas e apropriadas, utilizando bons fluxos de trabalho e uma ferramenta simples. O estudo procura investigar abordagens de base para uma boa gestão da informação num caminho acessível, com o objectivo de colmatar a lacuna de uma má estrutura de gestão da informação. O trabalho foi capaz de recolher a informação necessária através de entrevistas de investigação. Descobriu-se então que os sistemas de gestão de informação existentes no mercado são dispendiosos e complicados de migrar para a maioria dos casos. Inevitavelmente, tornando mais difícil para os estabelecimentos de pequena e média escala a adopção de recomendações de melhores práticas de normas internacionais. Este estudo analisa a gestão da informação relacionada com um importante contribuinte para o ciclo de vida do património construído que são os Arquitectos que fazem Arquitectura. O estudo centra-se numa prática de pequena escala com o objectivo de elevar o seu actual nível de maturidade, além de alinhar os actuais processos de informação com as normas internacionais relevantes. Contudo, existe uma maturidade significativa na gestão da informação utilizando a metodologia BIM dentro da empresa. O entendimento de que o berço da gestão da informação depende de fluxos de trabalho pragmáticos dos processos existentes que informam a melhoria proposta foi útil neste trabalho. A abordagem procurou ferramentas prontamente disponíveis e acessíveis que potencialmente melhorariam a gestão da informação em torno do ambiente de dados comum (CDE), convenções de nomenclatura, e verificação da informação (Baseado em pastas). Três soluções tentadas a partir dos estudos abrem a porta para mergulhar num amplo espectro de soluções baseadas numa síntese robusta de fluxos de trabalho pragmáticos em torno da prática da gestão da informação na Arquitectura e outros agentes da indústria utilizando a metodologia BIM. O trabalho não cria um CDE perfeito mas tentou desenvolver três (3) soluções semi-automatizadas de nível para ajudar a gestão da informação de uma forma estruturada. Finalmente, para criar estas soluções, foram implementadas três ferramentas de Tecnologia da Informação; linguagem de programação Python utilizada para o desenho lógico, Tkinter para python utilizado para a IU, e Microsoft Access para a Base de Dados.

Palavras-chave: (Gestão da Informação, Criador do nome do contentor, Metadados, CDE de base de dados, Verificação Automatizada)

ABSTRACT

Information Management using Building Information modelling (BIM) is a set of recommended concepts and principles to support the management and production of information during the life cycle of a built asset (EN ISO 19650-1, 2018). The management of Information is important, especially in recent times with tremendous forward push in digital transformation. The notion that information management is best done on robust information management systems (IMS) is correct. Fair to say that global momentum is going in the direction of cloud-based or electronic solutions but on the other hand many firms might be excluded because of market dynamics of affordability in relation to affordable capital investment at their disposal. Surely, it will come a time that the anticipated shift to these systems will happen, but the big question is what happens in the meantime? However, Information created, stored, used, and managed should be organised in a well-structured manner with the correct and appropriate recommendations using good workflows and simple tools. The study seeks to investigate baseline approaches to good information management in an affordable path, with the aim to bridge the gap of poor information management structure (in small and medium scale enterprises). The work was able to gather necessary information through investigative interviews. It was then discovered that information management systems out there in the market are expensive to afford and complicated to migrate to in most cases. Inevitably, making it harder for small and medium scale establishments from adopting best practice recommendations of international standards. This study looks at information management relating to an important contributor to the built asset life cycle which are Architects doing Architecture. The study focuses on a small-scale practice with the aim to elevate its current maturity level, plus aligning the current information management processes with relevant international standards. Though, there is significant maturity in information management using BIM methodology within the firm. The understanding that the cradle of information management relies on pragmatic workflows of existing processes that informs proposed improvement was helpful in this work. The approach looked at readily available and affordable tools that would potentially improve information management around common data environment (CDE), naming conventions, and verification of information (Folder-Based). Three attempted solutions from the studies opens the gate to dive into a broad spectrum of solutions based on robust synthesis of pragmatic workflows around information management practice in Architecture and other industry players using BIM methodology. The work does not create a perfect CDE but has attempted to develop three (3) Tier semi-automated solutions to assist the management of information in a structured way. Lastly, to create these solutions three Information Technology tools were deployed; Python programming language used for logic design, Tkinter for python used for the UI, and Microsoft Access for the Database.

Keywords: (Information Management, Container name, Metadata, Folder-base CDE, Automated Verification, SMEs)

TABLE OF CONTENTS

1. INTRODUCTION.....	13
1.1. MOTIVATION	14
1.2. PROBLEM STATEMENT	15
1.3. OBJECTIVES AND GOALS	16
1.4. DISSERTATION STRUCTURE	16
2. INFORMATION MANAGEMENT USING BIM	19
2.1. OVERVIEW CONCEPT OF INFORMATION MANAGEMENT.....	19
2.2. INFORMATION MANAGEMENT WITHIN THE ARCHITECTURAL REALM.....	20
2.3. INFORMATION MANAGEMENT, DIGITALIZATION, AEC INDUSTRY AND ISO 19650 1 & 2.....	22
2.4. STANDARDIZATIONS FOR INFORMATION MANAGEMENT. ISO 19650_1 & 2.....	23
2.5. INFORMATION MANAGEMENT SYSTEMS (IMS)	25
2.5.1. Electronic Document Management System application in the AEC industry.....	26
2.5.2. Components of Information Management systems.....	27
2.6. SALIENT POINTS OF MANAGING INFORMATION USING THE BIM METHODOLOGY.....	28
3. DIAGNOSIS OF CASE STUDY AND DEVELOPMENTS	31
3.1. OVERVIEW AND FOCUS AREA	31
3.2. INFORMATION MANAGEMENT WITHIN MARTA CAMPOS ATELIER DE ARQUITECTURA PRACTICE	31
3.3. INFORMATION MANAGEMENT WORKFLOWS (AS-IS).....	33
3.4. TABULAR SYNTHESIS OF WEAKNESS AND STRENGTHS.....	34
3.5. PROPOSED WORKFLOWS FOR IMPROVEMENT (TO-BE)	37
3.6. CONCEPTUAL ANALYSIS OF IMPROVEMENT AREAS	39
4. TECHNOLOGY SOLUTIONS (RESULTS)	43
4.1. DEVELOPMENT OF 3 TIER FOLDER BASED CDE KIT	43
4.2. CONCEPT PROCESS FLOW DIAGRAMS.....	44
4.3. TIER 1: SCRIPTED FOLDER BASED CDE CREATOR.....	45
4.3.1. Illustrative Structure	45
4.3.2. Graphical Interface User Instruction / Workflow Principle	45
4.4. TIER 2: SCRIPTED CONTAINER NAME CREATOR.....	46
4.4.1. Illustrative Structure	46
4.4.2. Graphical Interface User Instruction / Workflow Principle	47
4.5. TIER 3: AUTOMATED VERIFICATION SCRIPT	48
4.5.1. Illustrative Structure	48
4.5.2. Graphical Interface User Instruction / Workflow Principle	48

5. CONCLUSION	51
5.1. GENERAL CONCLUSION.....	51
5.2. RECOMMENDATIONS.....	52
REFERENCES	53
LIST OF ACRONYMS AND ABBREVIATIONS	55
APPENDICES	56

LIST OF FIGURES

Figure 1 - illustration showing basic components of information management (Mirus, 2017)	13
Figure 2 - Showing an AEC employment fragmentation.....	14
Figure 3 - illustrating information management through a coordinated workflow (van de Watt, 2022)	19
Figure 4 - historic progression of the Architects Role in information management (Marta Campos Atelier de Arquitectura, n.d.)	20
Figure 5- macro view of the communication process (Dawood et al., 2002).....	21
Figure 6 - RIBA plan of work (RIBA, 2020)	22
Figure 7 - macro view of the communication process (Saarikko et al., 2020).....	22
Figure 10 - PAS 1192 used as the basis for the ISO 19650 series. (shillcock & Operam, 2022)	23
Figure 11 - ISO 19650 series – part 6 is in development. (shillcock & Operam, 2022)	23
Figure 12 - illustration of how information system works.....	25
Figure 13 – worksheet used for assessment Source (BIM A+1).....	32
Figure 14 -Results from organisational BIM assessment.....	33
Figure 15 - Workflow (AS-IS) used for information management with internal and external CDE. ...	34
Figure 16 – Illustration of To-Be for the improvement of the current information management around the CDE (TO-BE).	38
Figure 17- CDE states for information management functions (ISO 19650).	39
Figure 18 - Proposed concept for generic CDE workflow using Folder based structure.	39
Figure 19 - ISO 19650-2 recommendation for information container naming (according to UK National annex).	40
Figure 20 - An example of a range of metadata that can be assigned in a cloud based CDE solution (Uk BIM Framework, 2021).	40
Figure 21 - An example of expanded range of metadata according to the need of Mc's Atelier.	40
Figure 22 - Illustration of two different CDE solutions where metadata assignment must transfer (UK BIM Framework, 2021).....	41
Figure 23 - Illustration of a workplan that would address the concept of the proposal.	41
Figure 24 - Diagrammatic representation of the solutions.	44
Figure 25 - Illustration of Folder based CDE, according to ISO 19650, which Includes expanded functions prescribed by MC's practice, Informed by their current practice.....	45
Figure 27 - Illustration of how information is verified	48

LIST OF TABLES

Table 1 - Extract from General knowledge around the components that make-up IMS	27
Table 2 - columnated identification of strengths and weakness.	34
Table 3 - Evaluation of existing information management in relation to key components of an Information management system.....	36

This page is intentionally left blank

1. INTRODUCTION

The emerging importance of Building Information Modelling (BIM) in Architecture, Engineering, and Construction (AEC) has accelerated the popularity and need for better information management in the industry. Information and management chained to form information management is a key ingredient of BIM. The ISO 19650-1 “sets out recommended concepts and principles for business processes across the life cycle of built assets (referred to as information management) when using building information modelling” (EN ISO 19650-1, 2018) this has placed information management in a strategic pivot position for BIM. BIM as a methodology has many definitions and dimensions as it applies to diverse types of assets. From the ISO 19650-1 document cited above, Information management is the center of whatever definition one might have of BIM as it relates to needs derived from it. BIM means different things to different people and in different contexts. (Demian & Walters, 2014).

The AEC industry can be attributed to Human evolution from time immemorial linked to the fact that shelter is a non-negotiable need of human beings. All through known memory in time past, various forms of information have been used and managed by people in all aspects of life, as crude as it might have been, information and management have existed. The general knowledge relating information and management to human and human activities, however, the level of sophistication, has indicated the tremendous need for valuable information management practice. Manual and digital Information management is primarily born out of a good culture meaning the practice of organization information management is rooted in human actions.

Another lens connecting information, collaboration and humans says “throughout the history of human civilization, specialization and division of labour have been key factors of progress. They were made possible by the cooperation of ever-larger groups of people. Construction has been no exception. However, specialization leads to fragmentation - unless specialization is supported by approaches and technologies that enable people to work together and that systems are interoperable” (Turk, 2020).

In a basic definition, Information management is the management of information, data, and the knowledge derived from it (Mirus, 2017). Its key components are Leadership, Operations, Processes, Data, and Technology. Figure 1 below explains from left to right the components. 1, The Leadership is the key decision maker that is on top of all strategic mandate. 2, Operations are people, project management, and communication when managing information. 3, Processes are workflows according to accepted and agreed standards. 4, Data is created, collected, organized, and maintained based on the first (1st) three components, and 5, Technology reflects efficient tools used to maintain and manipulate all other components.

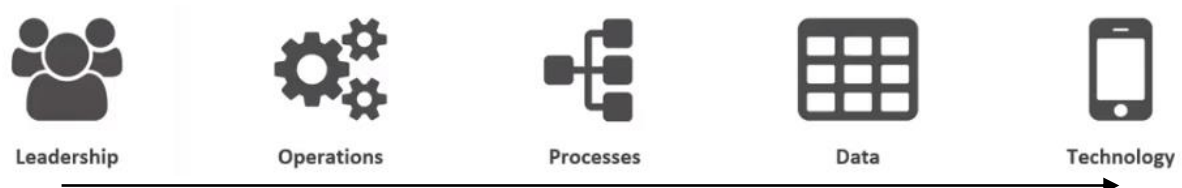
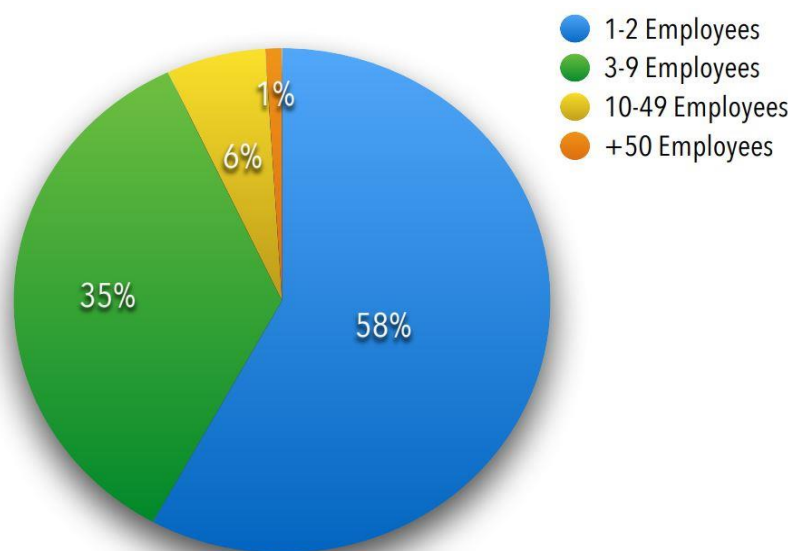


Figure 1 - Illustration showing basic components of information management (Mirus, 2017).

According to Werner, (2020) a renowned economist highlighted an interesting fact, “looking at the world today, two-thirds of employment is with small to medium scale enterprises, they account for most of the employment. According to International Labor Organization “There is broad agreement that small and medium-sized enterprises (SMEs) are vital to achieving decent and productive employment as they globally account for two-thirds of all jobs and create the majority of new jobs”. This phenomenon has an interesting connection to the subject matter “information management” in BIM, it shows that better information management in an affordable matter will expand the BIM methodology which would further elevate the industry (AEC). Hence, it can be said that improved information management could be a gateway to expansion for any establishment to gain more market share in its niche thereby impacting directly on the business be it BIM or any other sector.

Figure 2 below reinforces the importance of information management in the AEC industry. The pie chart diagram analysis shows that 58% Employees with the largest fragment account for most of the industry. Ultimately, if information management using BIM methodology would better the AEC industry ways of doing business if this sector has an affordable and simple means of managing information in the correct way and corresponds to relevant international standards.



Pazlar, T., Dolenc, M. in Duhovnik, J. 2004.
 Rezultati raziskave prodAEC o rabi informacijskih tehnologij v arhitekturi, inženirstvu in gradbeništvu v Sloveniji. *Gradb. vestn.* 53, 9: 223–229.

Figure 2 - Showing an AEC employment fragmentation.

1.1. Motivation

In theory, the environment used for information managed during the lifecycle of an asset should be in a common data environment management (CDE).

Though, there is a potential misconception that the CDE is more about technology and less about workflows.

Fundamentally, workflows are developed first, and solutions are selected to facilitate the workflow. It may also be understood that single technology solutions dominate project information management (UK BIM Framework, 2021). This is not the case, and many solutions exist to deal with diverse types of project information. There may, for example, be document management tools for design files, contract management tools that manage commercial information, email management tools for correspondence, and mobile-based tools for site quality data. Each solution may have multiple and different workflows ensuring that information is carefully planned, shared, stored, managed, and retrieved and that it is timely, correct, complete, and consistent (UK BIM Framework, 2021).

If a CDE is key to any project enabled with BIM to succeed, and the employment index globally shows that small and medium-scale firms and enterprises account for two-thirds of world employment. It will be fair to make the connection between small scale businesses and the need to improvement information management under the umbrella of BIM methodology and inter-relation with other sectors.

Besides, the work is carried out under the structure of a practising organisation with all the realities of the day-to-day execution of Building asset development from inception through to construction, eager to understand the challenges that come with translating frameworks, processes, standards, protocols, policies etc. to practical applications in terms of information management, particularly in the realm of Architecture and as it relates to allied Professionals using the BIM concept in small scale practice. Optimistic that what shall be realized will further improve the practice, align its operations to international standards, enhance internal processes, and further advance BIM adoption in smaller scale establishments.

1.2. Problem Statement

There is the dominant notion that information managed in BIM must be in an Electronic Document Management System (EDMS) explained as a robust software designed to centralize an organization's relevant documents and engineering drawings thereby replacing outdated disparate or manual systems. The goal is to provide a single and real source that facilitates collaboration, streamlines compliance, simplifies document version control, and derives great efficiency and ROI (enfoinformatics, n.d.). Attractive as it may sound, many firms cannot afford these solutions mostly because they might be more expensive. Fair to say that global momentum is going in the direction of cloud-based or electronic solutions but on the other hand many firms might be excluded because of market dynamics of affordability in relation to affordable capital investment at their disposal. Surely, it will come a time that the anticipated shift to these systems will happen, but the big question is what happens in the meantime? Should they stand aside and not partake in the good practice of information management using BIM? The alternative would be attempted in this work.

In the light of the above-mentioned, when EDMS is not used for the management of information certain drawbacks need to be addressed. Such as:

- 1, Information holding environments structured according to international standard recommendations (ISO – 19650) to support information management during the delivery phase.
- 2, Tackling the issue of proper naming convention that resides in the different states of the CDE.
- 3, Registry that keeps all information label with the appropriate identification code and metadata information in a sequential structure which is correct and with no duplicates.
- 4, The ability to automatically verify this information.

The above list of issues can be considered as the cradle of information management when not using an EDMS solution. Attempting to correctly do this will make a positive difference in the realm of any AEC industry player.

Limitations such as human working fatigue to follow through with all information management functions affects the organisation of information if manual process is used.

Certain functional requirements need to be smart, which requires a more sophisticated programming ability creates limitation to addressing some of the functions.

Time frame for the work has a direct impact on the overall aim.

1.3. Objectives and Goals

What we would like to achieve is a “CDE KIT” entangling attainable information management solutions, ISO-19650 focused, for small/medium companies to easily dive into information management in an ever more digitalised AEC industry.

This work aims at critical analysis of information management workflows (graphical and non-graphical) within the common data environment (CDE) as it is used in the establishment, as it relates to stipulations laid down in the ISO 19650-1&2 standards, as it connects to expanded functions (custom additions) and develops simple and affordable working scripts. within the realm of an Architectural Practice currently using BIM Methodology, (“Marta Campos – Arquitectura”) to identify weaknesses and strengths and proffer better workflows that comply with international standards.

Hence, improving and extending BIM methodology performance, elevate the current maturity of the practice through modified workflows based on ISO standards. The goal is to design technology solutions that create folder based CDE, container naming creators, and verification mechanisms for all information that will be in the CDE folders.

1.4. Dissertation Structure

Chapter 1 (**Introduction**) gives an elaborate explanation of different angles to the title and how these relate to one another and emphasises its importance. To talk about information management, it was good to link information and management as it has evolved through history (Human aspect). The sector involved in the use of information and its direct impact on employment was mentioned, and problems associated with excluding good practices due to economic situations followed, while the recommendation of the great standardization manual of BIM which is the ISO-19650 series, and its key connection to Information management was established. All this perspective feeds into the established problem manifest, extracted from the thoughts of the host organisation putting out the need to create useful information management practice that is affordable, though having its limitations does not warrant standing aside. Reflecting an important need for work in the niche.

Chapter 2 (**Information management in BIM and AEC**) The history of information management and the early proponents of this term in modern day highlight its main ingredients. The content is structured in 6 parts. **One**, BIM and the AEC relevant background work as it applies to information management (IM) currently ongoing and in the recent past. **Two**, the collected review works of IM in the realm of Architecture as it is a major part of the AEC industry with a focus on CDE. **Three**, ISO 19650-1,2,3 and 4 are currently part of the ISO 19650 series which is the leading standards that govern BIM implementation, careful extracts from this document provided the sole source of truth as to how this uniform way of doing things will eventually be a norm. This section will outline a digital pipeline supporting IM vis-a-viz ISO standards. **Four**, understanding management systems especially EDMS

which is highly recommended by most guidance documents, knowing how it works and understanding its back-end philosophy through existing literature available can give insights to creating plugin links to foster interoperability. **Five**, MC's information management knowledge area is well developed theoretically and in practice to noticeable maturity. The verbal narration of their management practice is captured in this section, with important insight coming from the leadership of the practice. A clear indication of strategies and the willingness to uphold them. **Six**, lastly important points of the ins and outs of information management in BIM are drawn and narrowed to Architectural practice and how it applies to allied professionals.

Chapter 3 (**Diagnosis of Case Study**) The period of the study was characterized by reoccurring meetings every other week. It was an open discussion that gave a deep understanding of the practice, its values, working methods, challenges, step-by-step approach to projects inclined to Architectural phases and delivery recommendations from the ISO 19650-2. Leading to an illustration of the process in an AS IS process diagram. Through this rigorous exercise, obvious strengths and weakness was picked on, which eventually helped to identify a course of action. Focusing particularly on the CDE and container naming convention. TO-BE proposed workflows of improvement built on the exiting status.

Chapter 4 (**Technology Solution**) Three Tier solution was considered, focusing on the CDE. Firstly, is the automated creation of folder-based CDE states recommended for information management over a project or an organized assets management procedure. Secondly, the creation of naming containers in a structured matter as stipulated by the UK National annex to the ISO-19650. Thirdly, is the verification of the information in the folders. As simple as the outcome of this approach sounds, it addresses important components of what characterizes the CDE. Good placeholders for different information stages, naming conventions and data repository that holds the correct metadata. A graphical user interface is developed for ease of use, connected to the database (Microsoft Access) through python logic language. Solutions demo illustrating diagrammatically proposed workflows.

Chapter 5 (**Conclusion/recommendations**) In summary outlines the results of this work and make a case for future works that could be done.

This page is intentionally left blank

2. INFORMATION MANAGEMENT USING BIM

Utilizing information management to manage the information about an asset throughout its life cycle, BIM primarily uses a three-dimensional (3D) model of the asset. The two things work best together. A good model is created by solid information management practices, and a good model effectively manages information.

2.1. Overview concept of Information management

Constructed from general readings, Information in general terms could mean the representation of messages of any form, it is not considered knowledge, rather it is an extract of data that is processed, organized, structured, and given a context to make it useful. To Put it in another form, Information simple or complex, structured, or unstructured is the fuel powering any organization for compliance, efficiency, effectiveness, etc. Strategic transformation depends on information without good information progress is uncertain. In general, management is the methodical process of controlling any kind of information within a structure by a series of actions and tactics that vary according to the type of thing that needs to be controlled. It is accurate to argue that something substantial has emerged and needs to be controlled in an organized manner when it comes to the topic of information management if the aforementioned explanations are combined to form information management. As shown in Figure 3 below, the CDE serves as the only source of truth for all information management activities that are strategically connected. The idea can be used in all areas of the AEC industry, as well as any other industry for that matter.



Figure 3 - illustrating information management through a coordinated workflow (van de Watt, 2022).

2.2. Information management within the Architectural Realm

Historically, information was managed by Architects as the sole controllers of the entire process. The dawn of the 20th century brought about releasing some of the Architect's information management power to different structures and fast forward to recent times the entire spectrum of the industry transformed into what is shown in figure 4 below progressively presenting different times.

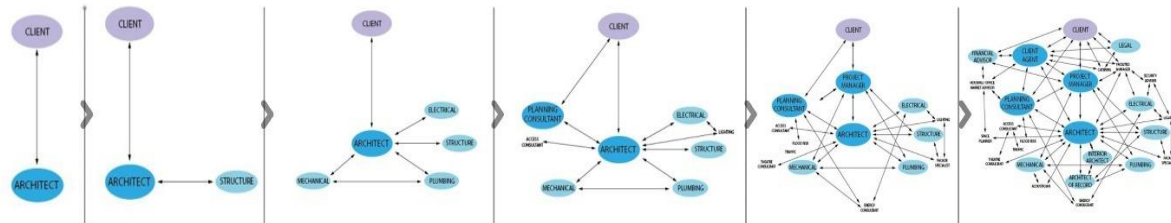


Figure 4 - historic progression of the Architects Role in information management (Marta Campos Atelier de Arquitectura, n.d.)

Figure 4 above with no doubt carries in its progressive pipeline journey through time. The illustration reveals what would be regarded as the roles of Architecture in the AEC industry across the timeline. On the contrary, embedded in it is the complexity and slow development of the magnitude of what it entails to be an Architect practicing Architecture. Generated in the flow of these matrices are containers of messages flowing in one direction, two directions, or multiple directions however the case may be, there is tremendous trouble created due to irregular patterns. Firstly, because there is no coherent manner in the way these messages are coordinated and managed.

The construction process generates a significant amount of information, much of which is not integrated but is instead created and held by various people and organizations, according to certain subsequent research in the connected domains of engineering and construction. This can create problems with keeping information up-to-date and ensuring that all parties requiring access to the information have the most recent version (Abukhder & Munns, 2003). Two-thirds of the construction problems are caused by inadequate coordination and inefficient means of communication of project information and data (Cornick & Barre, 1991). Integration of the design and construction processes alone cannot improve productivity and performance without improved communication and efficient means of exchanging information (Dawood et al., 2002).

Looking at another context, construction information can be grouped into three categories: general information, organization-specific information, and project-specific information (Dawood et al., 2002). Integration of the design and construction processes alone cannot improve productivity and performance without improved communication and efficient means of exchanging information. Fig. 5 shows a macro view of the current communication process in the industry. The same information, sometimes with different views, is often used in multiple functional areas of the construction process by multiple organizations involved in the project. The exchange of that information is still being carried out today in much the same way as it had been done in the last few decades-manually. In some

cases where computers are used to produce the data, they need to be reformatted before they can be imported and reused, resulting in misinterpretations or loss of vital information and errors (Dawood et al., 2002).

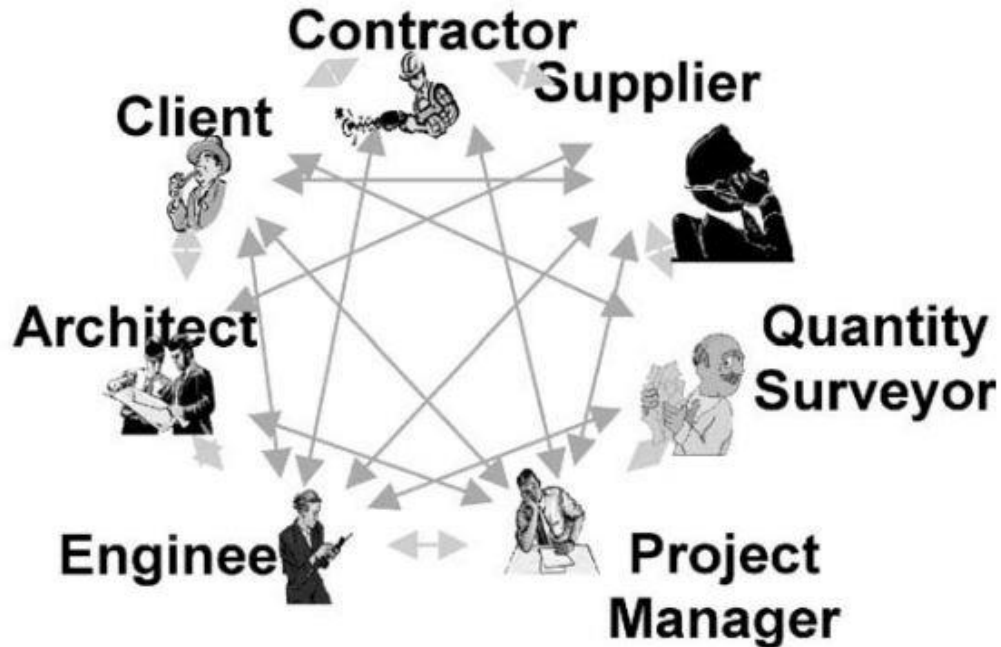


Figure 5- macro view of the communication process (Dawood et al., 2002).

In the business of Architecture and the interaction with clients, Engineers, contractors and so on, information is created which needs to be managed and managed properly is the main issue. That is where the break lies.

Figure 6 below, The Royal Institute of British Architects has a very robust plan of work which is used by organizations. It organizes the process of briefing, designing, constructing, and operating building projects into eight stages and explains the stage outcomes, core tasks and information exchange required at each stage. This table shows a comprehensive approach to how information potentially generated over a project or adopted by an organization is supposed to be coordinated.

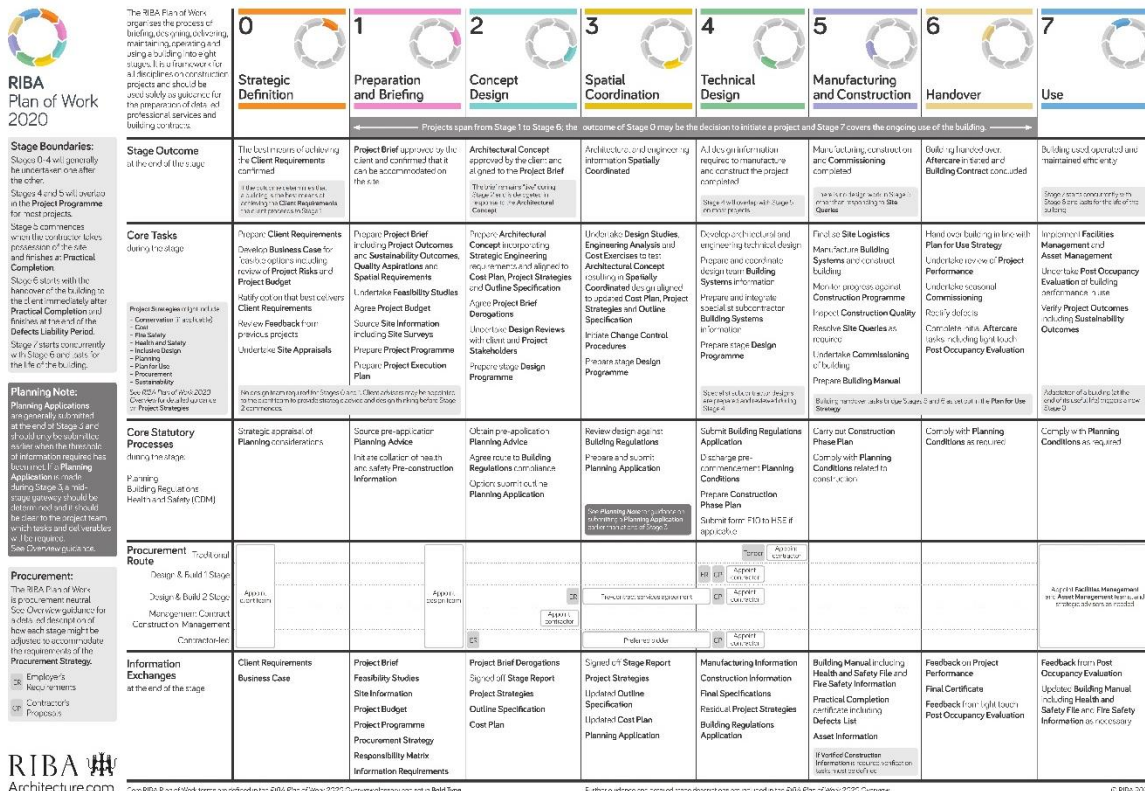


Figure 6 - RIBA plan of work (RIBA, 2020).

2.3. Information management, Digitalization, AEC industry and ISO 19650 1 & 2

Information management can be manual or electronic. So many factors affect transparency, efficiency, control, integrity, quantity, quality, and so on. It is no news that digital transformation has changed and is changing every aspect of today's world. Transformations come with so many perspectives though having a common baseline which in this case is technology. Figure 7 below at the apex, is digital transformation, followed by digitalization and digitization.

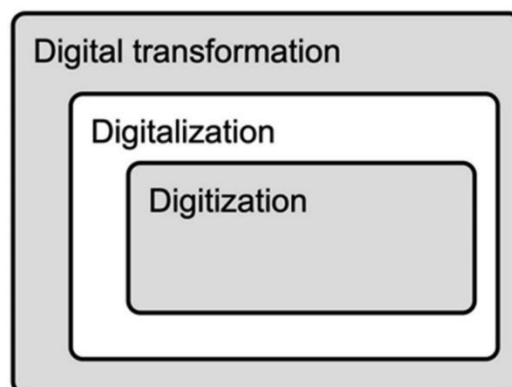


Figure 7 - macro view of the communication process (Saarikko et al., 2020).

digital transformation is the process of strategically reimagining and fundamentally overhauling how an organization uses people, processes, and technology—focusing on seamless integration of business

models and digital technology across all areas to enhance operations, bring better value to the customer, and drive revenue (Lukasa, n.d.)

Digitalization “The first contemporary use of the term “digitalization” in conjunction with computerization appeared in a 1971 essay first published in the North American Review. In it, Robert Wachal discusses the social implications of the “digitalization of society” in the context of considering objections to, and potentials for, computer-assisted human research. From this beginning, writing about digitalization has grown into a massive literature—one concerned less with the specific process of converting analogy data streams into digital bits or the specific affordances of digital media than with the ways that digital media variously structure, shape, and influence the contemporary world. In this sense, digitalization has come to broadly refer to the structuring of many and diverse domains of social life around digital communication and media infrastructures” (Brennen & Kreiss, 2016)

Digitization Scholars across disciplines use the term “digitization” to refer to the technical process of converting streams of analogue information into digital information (Brennen & Kreiss, 2016).

2.4. Standardizations for Information management. ISO 19650_1 & 2

Literature around digital transformation elaborated above shows the need to upgrade and have a seamless flow of information across the AEC industry. Analysis and facts presented show the potential of the expected growth. To achieve this goal of upgrading the AEC industry, standards such as the ISO 19650-series published by International Standards Organization was arranged to help the shift for Organizations towards digitization. Notable is the first series that includes information management using building information modelling

Placing importance on information management as the bedrock of digitization in the AEC industry. Key to all components, Architecture, Engineering, and Construction, with no one having less than the other in the level of cultivation towards a better system. It is worth listing out the ISO series as shown below in Figures 10 and 11 to give a background to its development.



Figure 10 - PAS 1192 used as the basis for the ISO 19650 series. (shillcock & Operam, 2022).



Figure 11 - ISO 19650 series – part 6 is in development. (shillcock & Operam, 2022).

The who, why, what is in it, and how can it help collaboration of information are explained below:

Who developed the ISO 19650-series? When the United Kingdom PAS 1192 (Publicly Available Specifications) series was published (28/02/2013), it was found that a lot of organizations around the world started to adopt the PAS 1192 series and make that requirement to their projects. Further to that the International Organization for Standards (ISO) through one of its committees (ISO/TC/059/SC/13) responsible for all standards around the Organization and digitization of information about building and civil engineering works including building information modelling (BIM) developed the ISO 19650 series (shillcock, 2022).

Why was it developed? Information has been produced in any case before the invention of any digital medium. Research had shown that information produced has been in silos and the training of creators of this information likewise. The impact of this is the production of wasteful information that invariably is incorrect, ambiguous, and incomplete resulting in significant waste in both cost and time. On the other side, with the development of different software solutions, to help with the problem based on promises made by vendors research has shown that it eventually gets worst with only the ability to produce inadequate quality information quicker than previously available methods. Tools are changing but the way we manage information is not. Hence, why the switch to collaborative ways of information management (shillcock, 2022).

What is in the ISO 19650 series? Part 1 **Concept and Principles**, for the management of information throughout the entire asset lifecycle that is both the delivery and the operation phase. Part 2 **Delivery Phase of Assets**, Sets out the information management process for the management of information during the delivery phase of assets. It includes a national annex. Part 3 **Operation Phase of Assets**, Set outs the information management process for the management information during the operational phase of the assets. It is a prequel to part 2. Asset owners should start from this point. Part 4 **Information Exchange** looks at the effect of exchange of information between parties. Though not published yet. Part 5 **Security Mind approach to information Management** Adopting security management approach to managing information. This came about because most asset owners' security policy is in chaos and now it is becoming structured and its online that comes with new treats and vulnerabilities, part of asset that are considered sensitive and helps to put measures in place to secure the associated information. Part 6 **Health and Safety** Work in Progress not published yet (shillcock, 2022).

The ISO standards is considered has one of the best baseline documentations used to achieve better information management within the AEC industry. it has disrupted the industry and will continue to change the way things are done as time will reveal. Many Governments are appending it into national requirements for certain works, Business, Organizations, and the academia are continuously combing it and testing it recommendations for new, existing, and experimental projects. In addition, Information management, Standards and Digital means is now fused together in a structured format to project meaningful collaboration and seamless linking of all parties in the AEC industry.

Another good explain of what the standards says about information management is found in the UK BIM alliance Guidance Documents. "Information management at its simplest is the management and execution of tasks relating to the definition of information requirements as well as information production, checking and delivery. It is fundamental to understand that information management applies to all information, collectively referred to as an "information model." An information model is a set of structured and/or unstructured information holding any combination of geometrical information, alphanumerical information, and documentation. Everyone making up a project team has an involvement and interaction with information management in varying degrees. It is not a new concept and has existed in some shape or form ever since information production was carried out by hand (UK BIM Framework, 2021).

2.5. Information Management Systems (IMS)

Architecture, Engineering, and Construction need to change, Standards are now available to guide everyone, and digital processes are highly sorted after to make it possible. Innovative ideas come with new challenges positive or negative. The information generated and managed needs mediums to be processed. Bringing into existence the need for proper Information manages systems.

An extract from (Bourgeois & Bourgeois, 2014) “Information systems (IS) is the study of complementary networks of hardware and software that people, and organizations use to collect, filter, process, create, and distribute data”.

“Information systems are combinations of hardware, software, and telecommunications networks that people build and use to collect, create, and distribute useful data, typically in organizational settings.”

“Information systems are interrelated components working together to collect, process, store, and disseminate information to support decision making, coordination, control, analysis, and visualization in an organization.”

As you can see, these definitions focus on two diverse ways of describing information systems: the components that make up an information system and the role that those components play in an organization (Bourgeois & Bourgeois, 2014). Figure 12 shows diagrammatically how information system works in an elaborate manner.

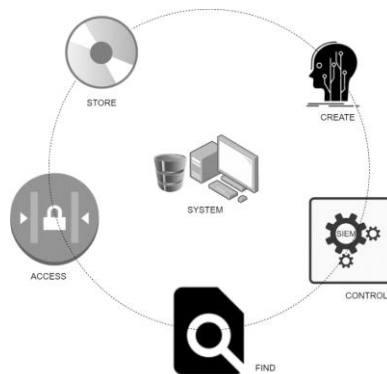


Figure 12 - illustration of how information system works.

Broadly speaking Management is the strategic coordination of any organization’s resources in a beautiful and systematic manner. Mostly with the intention to having better productivity that leads to better profits and allow for smoother continuity.

Now, this is to say that Information management systems is the fusion of all that is in management and information systems to makeup what we know as information management system. Information Management System (IMS) is a general term for software designed to help the storage, organization, and retrieval of information.

Furthermore, IMS is also the name of IBM’s mammoth software program developed in the 1960s to support NASA’s Apollo space program. This IMS version was the precursor to IBM’s premier hierarchical database management system (DBMS) (Techopedia, 2017).

2.5.1. Electronic Document Management System application in the AEC industry

Electronic document management system (EDMS) has been around for a while. Its rise can be traced back to the early days of the computer. EDMS is used by so many sectors to aid their various management processes. The new developments in the AEC industry have made it important to adopt EDMS as a better alternative to managing information contrary to the traditional way. An Electronic Document Management System (EDMS) is software that centrally stores and organizes documentation in one digital repository. Types of EDMS include ‘self-built,’ proprietary ‘on-premises,’ and cloud-based solutions. The aim of an EDMS is to bring structured and secure digital filing capabilities, discoverability, and control to all the documentation your business generates so it can function effectively. An EDMS acts as a sole source of truth that eases collaboration and saves your organization unnecessary costs (Techopedia, 2017).

How is EDMS linked to the AEC industry?

“The ISO 19650 Part 1 Concepts guidance available on the [UK BIM Framework](#) website explains how the Common Data Environment is a combination of technical solutions and process workflows.

A CDE solution could be software, or it could be another form of tool. If information is exchanged by a non-digital solution (for example, a postal service) and/or stored in an organized hard-copy cabinet (which may, for example, be needed on a sensitive project where digital methods are not allowed), then this can also be described as a CDE solution that can be supported by workflows.

It is more usual though, for digital solutions like **electronic document management systems (EDMS)** to play a big part in implementing CDE solutions and workflows. But it must be recognized that many different technologies can be used within a single workflow.

ISO 19650-2 envisages that a CDE is provided and managed by the appointing party (or a third party acting on their behalf), for the management of all information containers that are developed and exchanged with the appointing party throughout the life of the project from each delivery team. This is referred to in ISO 19650-2 as the project CDE.

However, ISO 19650-2 also envisages that delivery teams may implement their own (distributed) CDEs as well (but not instead of the project CDE). This guidance has examples of this scenario, which can introduce complexities into the management of information” (UK BIM Framework, 2021).

EDMS will be a major driving system if Architecture, Engineering, and Construction industry is going to upgrade in all aspects of the supply chain. Highlighted from the Extract above is CDE considered as the environment holding the sole source of truth for information that is to be managed across projects. EDMS can be local or cloud-based adopting this system for information management is currently more desirable. However, it can be a challenge to some establishments (especially small to medium companies) that wish to dive into managing information according to ISO-19650, if EDMS are the only reliable/robust and available solutions, because EDMS solutions tend to be paid information management platforms. This Argument is backed-up by the global employment index. That creates a huge challenge, how can this problem be approached is part of the big question. Around robust systems, especially BIM ways of doing business is an emerging frontier in its early days.

2.5.2. Components of Information Management systems

The components of management systems are important to how it is applied and why it is necessary. The ingredients outlined in the table 1 below describes each component. The components of a document management system are mirrored for the purpose of information management in a common data environment used in BIM enabled projects.

Table 1 - Extract from general knowledge around the components that make-up Document management system.

Source: (*Document management system*. Wikipedia. https://en.wikipedia.org/wiki/Document_management_system):

	Components	Description
1	Metadata	It is a piece of extended information attributed to each document the gives it a unique identification for potential inter-relation with other systems.
2	Integration	It is connected to the efficiency of the application program interface (API) that supplies management function that allows for integrated operations
3	Capture	Processing of Imaging functionality linked to optical recognition
4	Data validation	Rules for checking entry to guide data formats.
5	Indexing	Necessary for tracking, classification e.tc
6	Storage	Functions as a keeping place for information in a proper structure.
7	Retrieval	It is the pipeline for fetching information from a storage.
8	Distribution	it is key to collaboration in this case it supplies functionality for sharing and keeping original integrity.
9	Workflow	This function is one of the most essential functions because it shows exactly how things are to be done hence, foresters' continuity.
10	Collaboration	It is inherent to EDMS. The function that allows users and admins to work in cooperation.
11	Versioning	Processes will always meet changes. This function helps to keep the old and new in an

		orderly sequence.
12	Searching	Systems are always interrogated through searching.
13	Federated search	The capability to coordinate individual sets of works from many sources with one system. This function enables a holistic check for error primarily.
14	Publishing	Indicates the mechanism for publishing outcomes of processes in different formats
15	Hardcopy reproduction	A tangible representation of document outputs.

2.6. Salient points of managing information using the BIM methodology

One, Building Information Modelling (BIM) is an expanding collection of concepts and tools which have been attributed with transformative capabilities within the Architecture, Engineering, Construction, and Operations (AECO) industry (Succar, 2010). Architects are vital contributors to AEC industry. The complexity and meticulous implementation of BIM comes with enormous information management requirement. Hence, creating the necessity for good information management practice with nuclear parties.

Two, in recent years, the importance of useful information management with documented responsibility and accountability for the activities and tasks involved has become more and more important. A proper focus on information management can help to achieve better project outcomes through the following:

- Improved coordination and communication.
- Better quality information production.
- Timely information delivery.
- Mitigated rework, unnecessary waste, and cost.
- Effectively informed decision making.
- More accurate audit trails/record.

These enablers demand collaborative working and are reliant on the effective management of information the whole life of assets to help improve our built environment.

The necessity for valuable information management is heightened further with the move to digital ways of working. To unlock efficiencies and effective use and reuse of information we need to specify, produce, check, approve and exchange information in a consistent and structured manner. While the use of technology brings many benefits, it relies on these consistent and structured approaches. The risks generated by poor procurement and management of information delivered digitally could be considered greater than its analogy predecessor via, for example, inappropriate access or distribution of information and security breaches.

It is essential that the industry focuses on “information as an asset” as much as the physical asset itself. At both a business and project level, outcomes are determined by decisions, and decisions are based on information. We, therefore, need to make sure that information management is embedded and treated with the same importance as design, project, and asset management (UK BIM Framework, 2021)

Three, Studies have shown that the AEC industry in some sectors has low digitization than others. The disparity amongst the various groups has tremendously slowed the overall growth potentials of the industry. What it means is that the lifecycle flow across the industry that is supposed to be seamless is leaking or broken along the path or at transition points. This places great emphasis on the need to standardize and manage information in a very structured manner so that flow of information facilitated by information technology can push digital and operational transformation in the AEC industry to the next level and on a broader coverage in respect to economic strength and participation.

This page is intentionally left blank

3. DIAGNOSIS OF CASE STUDY AND DEVELOPMENTS

3.1. Overview and focus area

Information management organized around CDE/containers, design objects and design materials was presented as the areas that need improvement. The study focusses on the CDE, with the preposition that why should small and medium scale establishments loss the chance to adopt and transform their business processes, simply because solutions provided by industry vendors are mostly expensive and not so economically viable. The hypothesis suggests a thorough investigation of existing maturity level within the practice that will inform ways to develop workflows and simple technology solutions that is not tedious to manage and that prepares the path for migration to desired system when need arises.

3.2. Information Management within Marta Campos Atelier de Arquitectura practice

Marta Campos Atelier de Arquitectura is a unique architectural practice with a keen inclination to BIM working methods. The practice has significant maturity in using BIM according to ISO 19650 recommendations. The leadership of the atelier is very aware of the pros and cons of using BIM in architectural practice with a suitable number of practice years. However, BIM organizational assessment tool was injected to ascertain maturity level.

The information management practice of the atelier intersects with delivery phases recommended by the ISO-19650. A critical mapping of architectural phases based on the RIBA plan of work, PIM Process according to ISO-19650-2, deliverables that the ISO 19650 recommends, and current organizational practice that aligns with the local regulations forms what is used to manage projects

The practice interlaces with allied professional in divergent phases of their current practice and directly with many clients. The Information management process currently is at a proficient level. However, good can be better and continues reinventing once self in an ever-competing market especially for small and medium scale organizations is important. Nevertheless, the leadership is very inclined to BIM and implements it on projects as an internal mandate to add values to their works.

Figure 13 below shows the worksheet used to generate the diagram used below. It is a self-assessment tool whereby, Inputs are selected in accordance with a corresponding level of maturity, based on planning elements, followed by indications of levels of maturity from 0 – 5(incrementally). 0 as non-existing and 5 as optimal. The colour deep magenta is used to indicate current level and deep maroon is used to indicate target level. Overall, the five sections namely, strategy, BIM Uses, Process, Information, Infrastructure, Personnel has an extended perspective of the practices BIM maturity.

Planning Element	Description	Level of Maturity						Current Level	Target Level	Total Possible
Strategy		0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	6	11	30
Organizational Mission and Goals	A mission is the fundamental purpose for existence of an organization. Goals are specific aims which the organization wishes to accomplish. A vision is a picture of what an organization is striving to become. Objectives are specific tasks or steps that when accomplished move the organization toward their goals.	No organizational mission or goals	Basic organizational mission established	Established basic organizational goals	Organization mission which addressed purpose, services, values (at a minimum)	Goals are specific, measurable, attainable, relevant, and timely	Mission and goals are regularly revised, maintained and updated (as necessary)	1	2	5
BIM Vision and Objectives		No BIM vision or objectives defined	Basic BIM vision is establish	Established Basic BIM Objectives	BIM Vision address mission, strategy, and culture	BIM objectives are specific, measurable, attainable, relevant, and timely	Vision and objectives are regularly revised, maintained and updated (as necessary)	1	2	5
Management Support	To what level does management support the BIM Planning Process	No management support	Limited support for feasibility study	Full Support for BIM Implementation with some resource commitment	Full support for BIM Implementation with appropriate resource commitment	Limited support for continuing efforts with a limited budget	Full support of continuing efforts	2	5	10
BIM Champion	A BIM Champion is a person who is technically skilled and motivated to guide an organization to improve their processes by pushing adoption, managing resistance to change and ensuring implementation of BIM	No BIM Champion	BIM Champion identified but limited time committed to BIM initiative	BIM Champion with adequate time commitment	Multiple BIM Champions with each working Group	Executive Level BIM Support Champion with limit time commitment	Executive-level BIM Champion working closely with working group champion	2	2	10
BIM Planning Committee	The BIM Planning Committee is responsible for developing the BIM strategy of the organization	No BIM Planning Committee established	Small Ad-hoc Committee with only those interested in BIM	BIM Committee is formalized but not inclusive of all operating units	Multi-disciplinary BIM Planning Committee established with members from all operative units	Planning Committee includes members for all level of the organization including executives	BIM Planning decisions are integrated with organizational Strategic Planning	0	0	0
BIM Uses	The specific methods of implementing BIM	0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	3	7	15
Project Uses	The specific methods of implementing BIM on projects	No BIM Uses for Projects identified	Minimal owner requirements for BIM	Minimal BIM Uses required	Extensive use of BIM with limited sharing between parties	Extensive use of BIM with sharing between parties within project phase	Open sharing of BIM data across all parties and project phases	2	5	10
Operational Uses	The specific methods of implementing BIM within the organization	No BIM Uses for Operations identified	Record (As-Built) BIM model received by operations	Record BIM data imported or referenced for operational uses	BIM data manually maintained for operational uses	BIM data is directly integrated with operational systems	BIM data maintained with operational systems in Real-time	1	2	5
Process	The means by which the BIM Uses are accomplished	0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	2	8	10
Project Processes	The documentation of External Project BIM Processes	No external project BIM processes documented	High-level BIM process documented for each party	Integrated high-level BIM process documented	Detailed BIM process documented for primary BIM Uses	Detailed BIM process documented for all BIM Uses	Detailed BIM process documented and regularly maintained and updated	1	5	5
Organizational Processes	The documentation of Internal Organizational BIM Processes	No Internal organizational BIM processes documented	High-Level BIM process documented for each operating unit	Integrated High level organizational process documented	Detailed BIM process documented for primary organizational Uses	Detailed BIM process documented for all BIM Uses	Detailed BIM Process documented and regularly maintained and updated	1	3	5
Planning Element	Description	Level of Maturity						Current Level	Target Level	Total Possible
Information		0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	6	10	30
Model Element Breakdown (MEB)	Model Element Breakdown Structure are identifiers assigned to each physical or functional element in the breakdown of the facility model.	No consistent Organizational Model Element Breakdown	Organizational Model Element Breakdown defined but not uniform within entire organization	Organizational Model Element Breakdown is uniform within the organization	Organizational Model Element Breakdown aligned with industry standards	Organizational Model Element Breakdown updated along with industry standards	Organizational modifications to industry standard model element breakdown are balloted for inclusion in industry standards	2	4	10
Level of Development (LOD)	The Level of Development (LOD) describes the level of completeness to which a Model Element developed	No consistent Level of Development	LOD defined but not standardized with the entire organization	LOD standardized within the organization	Organizational LOD standards aligned with industry standards	Model View Definitions & Information Delivery Manuals are used to define LOD	Organizational modification to MVDs and IDs are balloted for inclusion in industry standards	2	3	10
Facility Data	Facility Data is non-graphical information that can be attached to objects within the Model that defines various characteristics of the object	No consistent facility data requirement	Facility data defined but not internally standardized	Facility data defined and standardized within the organization	Organizational facility data attributes aligned with industry standards	Facility data attributes aligned with open standards	Facility data attributes updated with open standards	2	3	10
Infrastructure	Technological and physical systems needed for the operation of BIM with the organization.	0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	7	11	35
Software	the programs and other operating information used by a computer to implement BIM	No BIM Software	Software capable of accepting BIM data	Facility data defined and standardized within the organization	Advanced BIM software systems	All software systems available to all personnel	Program established for continuous updating of BIM software systems	4	5	20
Hardware	physical interconnections and devices required to store and execute (or run) BIM software	No Hardware capable of running BIM software	Some hardware capable of running basic BIM software	Facility data defined and standardized within the organization	Some advanced hardware systems with the organization	All organization hardware is capable of running advanced BIM Software	Program established for continuous updating of BIM hardware systems	2	3	10
Physical Spaces	Functional areas within a facility used to properly implement BIM within the organization	No dedicated BIM space	Single workstation for viewing BIM data	Facility data defined and standardized within the organization	BIM room for collaborating with large screen viewing capability	Multiple collaborative workspaces within regular workspace	Program established for continuous updating of BIM spaces	1	3	5
Personnel	Human resources of an organization	0 Non-Existent	1 Initial	2 Managed	3 Defined	4 Quantitatively Managed	5 Optimizing	7	10	35
Roles and Responsibilities	Roles are the primary function assumed by a person within the organization and Responsibilities are the tasks or obligations that one is required to do as part of that role.	No roles and responsibilities documented	BIM is the responsibility of the BIM Champion	BIM is the responsibility of the interdisciplinary BIM Group	BIM responsibility lies with each operating unit	BIM responsibility lies with each person	BIM Responsibilities are regularly reviewed to ensure they are properly distributed	3	5	15
Organizational Hierarchy	An arrangement of personnel and group into functional groups within the organization	Organizational Hierarchy does not address BIM	BIM Champion outside of typical organizational hierarchy	Small BIM Implementation Team outside the typical organization hierarchy	Large interdisciplinary BIM Group created	BIM Champion defined within each operating unit	BIM Implementation Team support BIM Use within operating units	2	0	5
Education	Education is to formally instruct about a subject	No Education Program	Ad hoc education as needed	Formal Presentations on what is BIM and the Benefits is has for the organization	Regularly conducted employee education sessions	On-Demand education program established for the organization	Education is seamlessly improved through lessons learned within the organization	0	0	5
Training	Train is to teach so as to make fit, qualified, or proficient in a specific task or process	No Training Program	Training program run by vendors - only for necessary personnel	Internal Training program for all personnel that may interact with BIM	Regularly conducted and routine training programs	On-Demand training program established for the organization	Training is seamlessly improved through lessons learned within the organization	0	0	5
Change Readiness	The willingness and state preparedness of an organization to integrate BIM	No Change Readiness Awareness	Established need for BIM	Upper management buy-in	Operating unit buy-in	All individuals buy-in	Willingness to change is part of the culture of the organization	2	5	5
Totals	This is the total for all the categories. Note this does reflect maturity in all sections. While the organization could score high, there could be some key areas not implemented that could hinder the organizations BIM Implementation.							29	49	145

Figure 13 – worksheet used for assessment Source (BIM A+1).

Based on the information gotten from series of interview with the BIM champion within the office, the information was injected into the matrix document for organizational BIM assessment, that covers a series of key planning elements inclusive of “process and information” relevant to the study. It revealed strong desire to improve processes. Figure 14 below shows that overall, BIM maturity level for an establishment of its size is significant and has a strong desire to grow more into using BIM process.

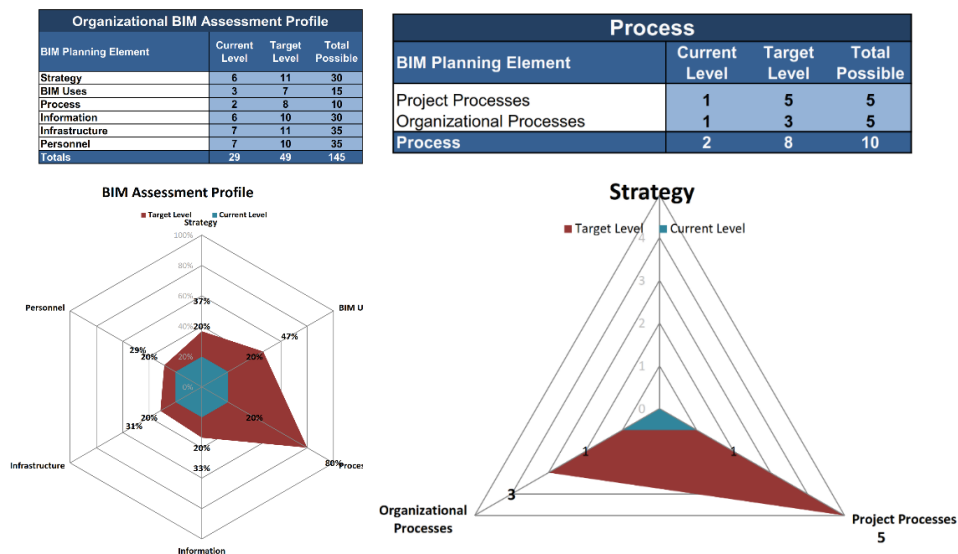


Figure 14 -Results from organisational BIM assessment.

3.3. Information management workflows (AS-IS)

The information management workflow in the practice is done internally with team members on a folder based CDE in their internal server while communicating with external professionals using Dropbox. Figure 19 Below diagrammatically shows the information management process over a project within the practice. The practice uses affordable resources to coordinate, share, manage, retrieve, and store information on a project-to-project bases. A parent folder holder is used to house two main folders, namely internal and external. Internal folder structure keeps information shared by local team working on a project. External refers to sets of information agreed and approved for sharing with specialist teams such as Electrical Engineers, Mechanical Engineers, structural Engineers etc. Manual transfer of information is used to shift information across the platforms. Furthermore, a crucial entity for identifying information in both structures is the container naming convention from the UK national annex. The process model in Figure 15 below, illustrates the information flow as it exists within the architectural practice of MC. When a project is initiated by a client who can be an individual, organisations, or government, that starts the cycle of an asset. The process begins to create information between the owner (Appointing Party) and the Architects (Lead Appointed). It corresponds to the Architectural phase 0_strategic definition and so on. Series of activities happen in this section and next stage can only continue if there is full agreement and approval to go ahead. From left to right the stages apply along the column while from top to bottom the process flow and a ridge connection to serve as demarcation.

Information generated needs to be managed from the moment it is created. The status according to the illustration below takes place on two fronts, on a Windows operating system folder structure (Internal) and cloud-based (External). The information managed is in different forms and formats.

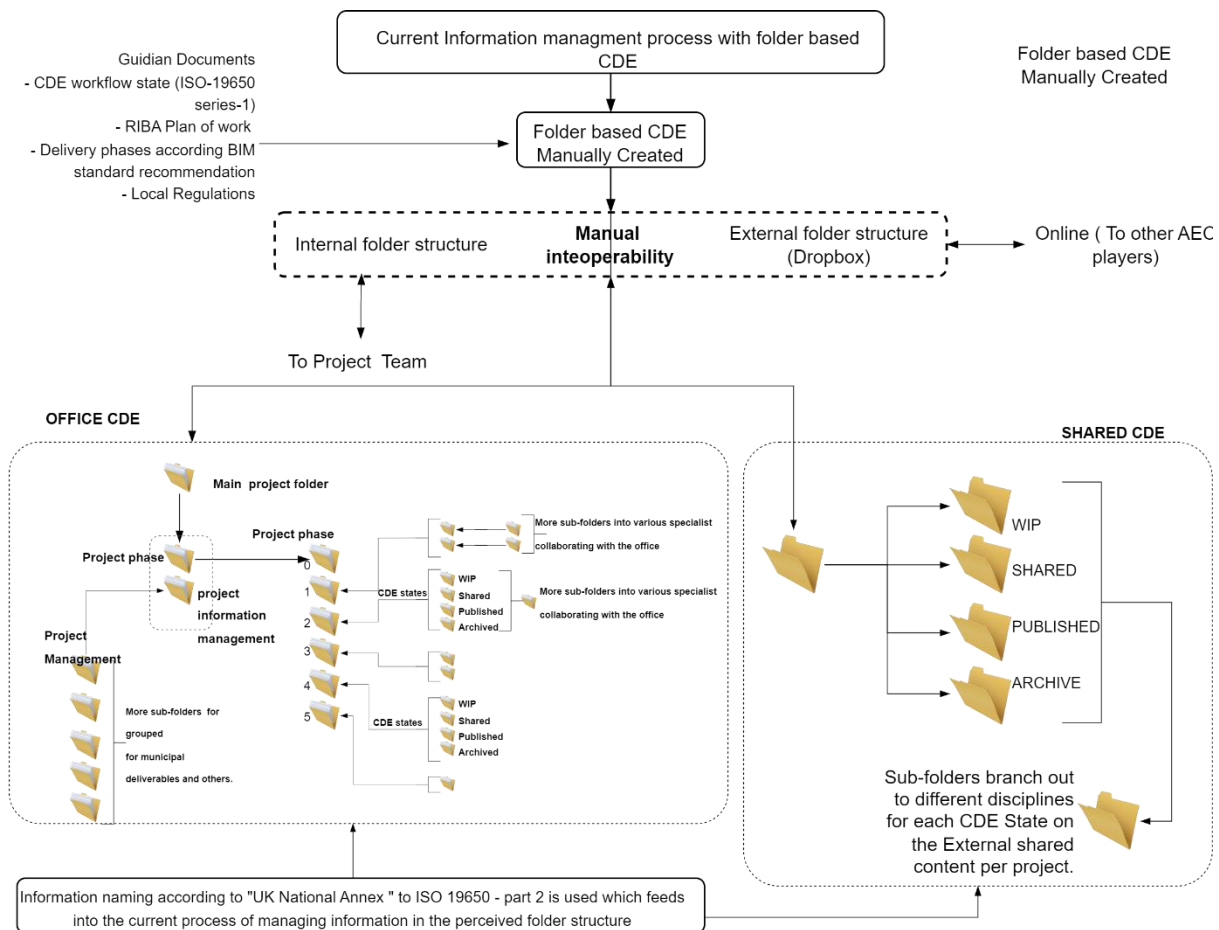


Figure 15 - Workflow (AS-IS) used for information management with internal and external CDE.

3.4. Tabular synthesis of weakness and strengths

Sorted out in Table 2 below shows identified strengths and weaknesses extracted during Interviews and periodic meetings during the study period. Narrows down to the following:

Table 2 - columnated identification of strengths and weakness.

Strengths	Weakness
1 The use of international standards and globally recognized guidance documents. Namely:ISO-19650, RIBA plan of work, and UK BIM frame.	The information management system around the CDE is currently organized differently, within Project team and other AEC players. Hence, making the flow of information difficult.
2 For information management best practice to really align to any practice and succeed the most critical component is the leaderships willingness. In this case the leader of the	For BIM enabled projects to be interoperable from one platform to another good metadata structure is to be in place. So that mapping or migration of information would be seamless.

practice is the captain steering the Voyage.

3	Modelling using computer-aided software is at an advance level plus information mapping on modelled elements and components is meticulously done.	Information management is steered by people and their willingness to follow it through as agreed. To assist this commitment, database structure manipulations is directly connected to it. In this case of study there are limitation in database management of information produced across a project.
4	The scale of the establishment is an advantage in the sense that People management and technology skill upgrade might be easily embarked upon as opposed to larger scale establishment.	Naming convention has a significant role in the organization of information. There is a limitation to the manual creation and storing the sequence of created naming formats. Tendency for error, fatigue as the project scales up are some of the flaws with the current approach.
5		Lack of storage history for container names. The tendency for information to be lost is high and traceability for projects that are for long periods becomes difficult.
6		Lack of storage for metadata associated to the information containers.
7		Lack of interoperability with other systems.

Table 3 - Evaluation of existing information management in relation to key components of an Information management system

The rating parameter ranges from 0-5. Zero (0) as the non-existing and Five (5) as Excellent. One (1) is the baseline for measurement and level of improvement increasing in positive multiples. Nine (6) out of Fifteen (15) components are low. There is awareness about the need to have but no implementation. In the middle, ratings 2-3 indicates components implemented that needs improvement. Overall, based on the IMS need to have guiding components the Information Management mechanism of the Atelier is low.

3.5. Proposed workflows for improvement (TO-BE)

After many deliberations, searching and researching on existing knowledge about common data environment. The approach mechanism employed, saw the possibility of organising information that resides in a CDE with readily opensource platforms available.

The major problems of the folder based CDE solutions that is used in the practice are:

- 1, Lack of automated creation of parent folders and sub-folders. Hence, keeping consistency and traceability difficult.
- 2, Non-Existence of access control.
- 3, Lack of proper storage for information created and used within the CDE.
- 4, Non-Existence of metadata mapping registry.
- 5, Lack of revision control.
- 6, Manual interoperability of information between the office and shared CDE.
- 7, Naming convention creator is rigid and prone to errors with multiple users.

Improvement around automated folder creation, automating naming creation, and verification of information as explained below, seeks to positively advance the current information management in the practice:

- 1, Design of an automated folder creation tool. Serving as a generic tool that is used from project to project. Tackling errors and inconsistency.
- 2, Generating automatic container naming identifications (ID) for all information. keeping an organised record of IDs created with all relevant information including the Metadata.
- 3, All information created, used, and stored need to be verified. Hence, a verification routine defined by a tool designed to check and validate information using the ID structure as a checking tool.

Considering that simple and affordable solutions is created with available tools, creating some level of automation so that errors are minimised, accuracy is assured, reduction in process fatigue, making sure workflows are easy to follow, creating the appropriate metadata structure that shall provide easy interoperability with other systems and verification of information (flagging off inconsistencies).

Figure 16 below shows the re-engineered process (TO-BE) with 3 three tiers semi-automated application all working to better arrange information and manage the information created, used, stored, and archived. Using a database platform for information safe keep, easy to use interface and connected with a logic language mechanism.

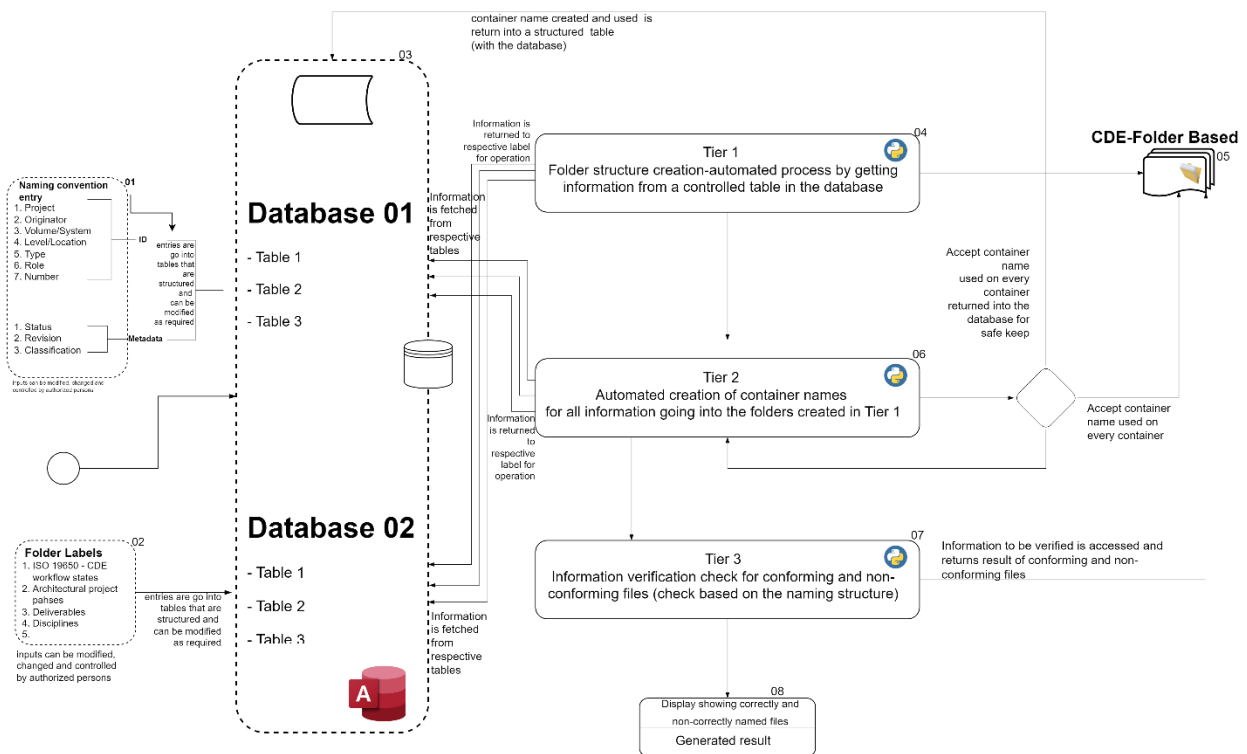
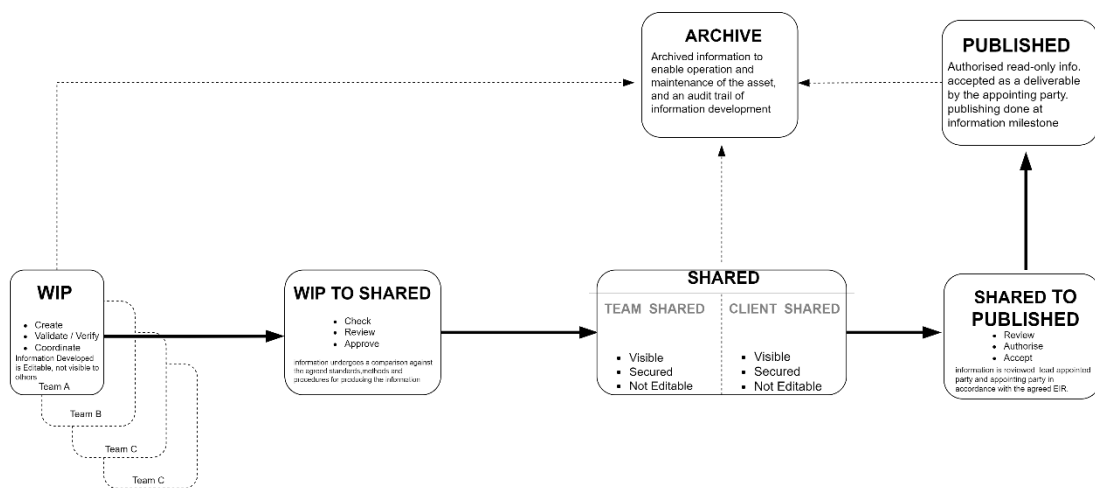


Figure 16 – Illustration of To-Be for the improvement of the current information management around the CDE (TO-BE).

3.6. Conceptual analysis of improvement areas

- 1 The CDE template is to be a Folder based structure developed around the states and the transition path for the states (placeholder) for information that would be created and managed during the life cycle of an asset. Respective folders which are simply created with any computer system will represent the states (it can be done manually or automated) work in progress/shared/archive/published. In addition to following what was stipulated by international standard, expansions to include other local requirements is formed in the structure such that the entire folder project is keep organised for information management. (Which includes, querying, auditing, access etc.) Figure 17 shows the concept as prescribed by ISO 19650 while figure 18 shows the concept adjusted to fit to purpose for the case study.



CDE workflow illustrated as specified in ISO 19650 -1 and 19650 2

Figure 17- CDE states for information management functions (ISO 19650).

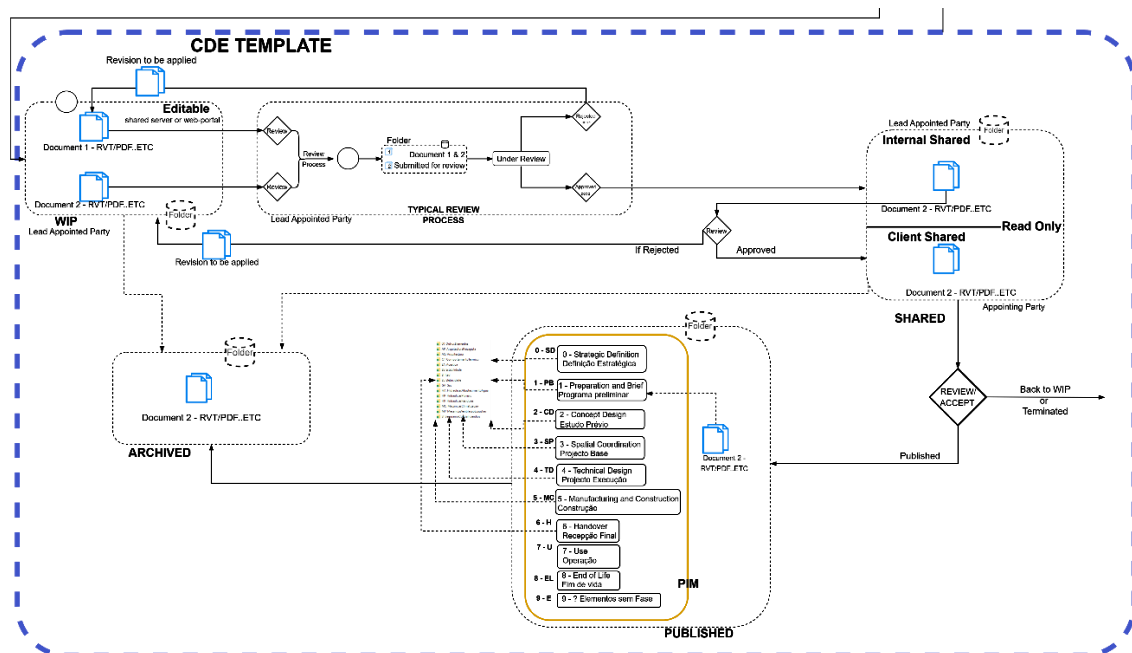
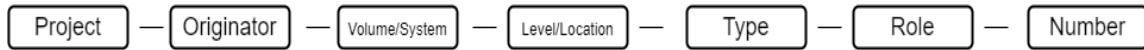


Figure 18 - Proposed concept for generic CDE workflow using Folder based structure.

- 2 Creating Container Names has to do with naming convention that is usually a mandatory agreement between parties involved in using BIM. In this case the Architectural Practice adopts the UK National Annex to BS EN ISO 19650-2:2018 (figure 19) for the purpose of Information container Identification. All information that shall and would be within an agreed CDE no matter the platform, a careful and well-structured information container is a need to have so that the management aspect of the information is easy and the potential benefit for interoperability is possible and much more.

Unique ID



Meta-Data

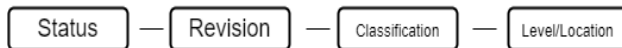


Figure 19 - ISO 19650-2 recommendation for information container naming (according to UK National annex).

In the concepts and principles document (part 1) naming convention is to be agreed on from the on-set, just like the annexing document suggested though it is not limited only to what is recommend. It can be expanded to have additional fields as the case maybe depending on a project or organizational requirement. Examples of case scenarios is shown in figure 20 as recommended by UK BIM guidance document.

Container Name	Description	Status	Revision	Author	Submittal Date	Container Classification
7001-BBH-ZZ-ZZ-DR-A-00301	First Floor Plan	S3	P04	Joe Blogs	12/11/2017	PM_40_30 : Design information
7001-BBH-ZZ-ZZ-DR-A-00312	West Elevation	A3	C06	Joe Blogs	12/11/2017	PM_40_30 : Design information

↑ Container Name / ID Field

↑ Additional Container Metadata Assignments

Figure 20 - An example of a range of metadata that can be assigned in a cloud based CDE solution (Uk BIM Framework, 2021).

Likewise in MC's practice it adopts the requirement of the UK National annex with additional information fields of its own as shown in figure 21 below:

CONTAINER NAME							METADATA		METADATA (OUTROS)					
UK	Project	Originator	Volume/System	Level/Location	Type	Role	Number	Status	Revision/Classification (Uniclass2015)	State	Author	Submittal Date		
IDENTIFICAÇÃO DO FICHEIRO							METADATA		METADATA (OUTROS)					
PT	Projecto	Origem	Volume/Sistema	Nível/Piso	Tipo	Disciplina/Função	Número	ID Ficheiro	Status	Revisão	Classificação	Estado	Autor	Data de submissão
	P2002Massenet-	MOOB-	IX-	05-	SS-	HP-	100000-	P2002Massenet-MOOB-IX-05-SS-HP-100000-S1-P01.1	S1-	P01.1	FI_60_53	SHARED	Paulo Monteiro	2022-02-15

Figure 21 - An example of expanded range of metadata according to the need of Mc's Atelier.

- 3 Verification of information has to do with checking for conforming and non-conforming information container within the CDE states so that regular assurance is ascertained time to time over the course of the project. Consistency of information is key for seamless inter-relationship of information across systems of all parties involved. Sharing and exchanging information depends on good information management structure. Key

ingredient of that is verification routines assisting in checking that information is managed properly.

- 4 Realistically depending on involving parties on a project there might be pockets of CDE's created that need to be interoperable with one another. Building information management in the manner so far recommend by standards and all supporting methodologies makes this process possible one. As shown below in figure 22 the interoperation between two CDE solutions. Interoperability of systems is the lifeline of collaborative work.

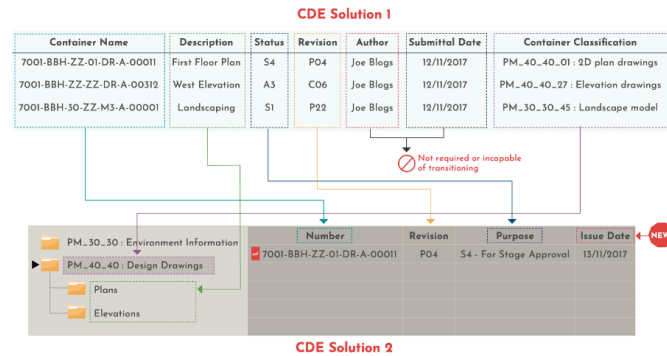


Figure 22 - Illustration of two different CDE solutions where metadata assignment must transfer (UK BIM Framework, 2021).

The Proposal (Figure 23) coined around four steps aims to proffer these solutions for the practice to improve the current way they manage information internally and externally in an inexpensive way. Whereby, information management around the main need to have in a CDE and the containers that are kept within the CDE is properly managed in accordance with good international practice. Making it possible with simple and free to use tools at disposal. The proposal looks at automating the process to a large extent thereby tackling issues of errors and reduction of workplace fatigue generated because of repetitive task. The workplan strategy is captured in Figure 31 below.

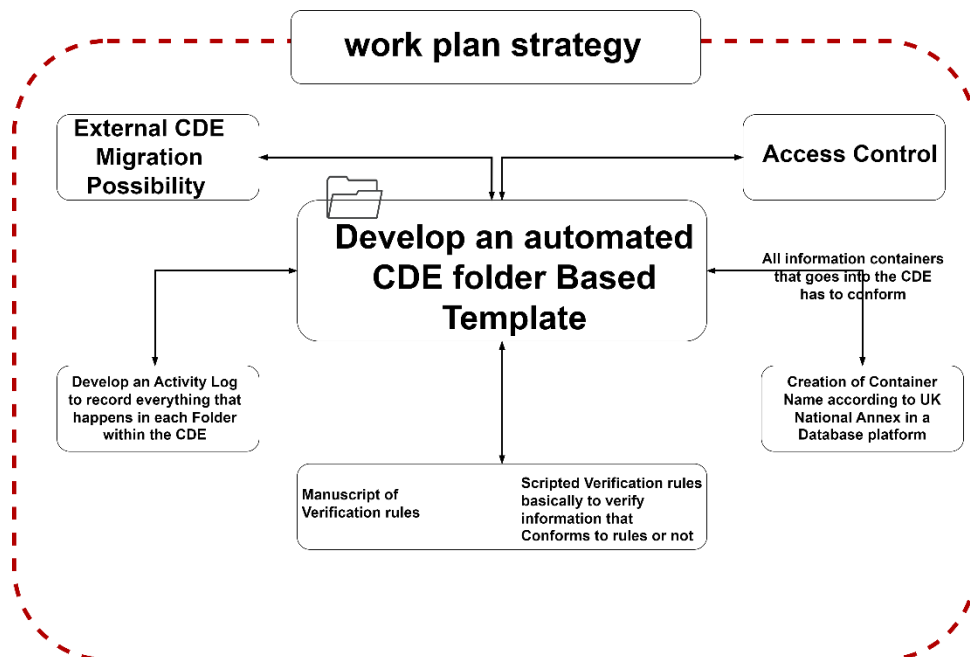


Figure 23 - Illustration of a workplan that would address the concept of the proposal.

This page is intentionally left blank

4. TECHNOLOGY SOLUTIONS (RESULTS)

4.1. Development of 3 Tier Folder based CDE KIT

A kit is a set of accessories used for a specific purpose. The solution is classified into three Tiers 1, 2 and 3. Folder creation, Naming convention creator and verification routine.

First Tier is the development of folder based CDE structure that answers the recommendation of ISO 19650, alongside the internal requirements of the practice. This solution is designed to be as generic as possible. The flexibility will make it expandable across different project types and requirements as the case maybe. A Database structure was developed in Microsoft Access (familiar and used by MC's practice) according to the proposed folder structure. Building the structure in a database makes information mapping correct and accurate from day one, a much-needed approach of information management good practice. Thereafter, A logic in this case python programming language is used to create an intelligent logic to perform two key functions. One, to query the database and fetch all information need to label each folder correctly. Two, is to create the structure according to the selections made on the Graphical User Interface (GUI) automatically to a directorate of the parent folder.

Second Tier is the creation of container naming Identification (ID). As an improvement on the current method of creating ID's which uses the Microsoft excel platform. A database is used to migrate all information from the excel platform so that consistency of the structure of information creation would be achieved and scalability would be managed better. Considering that reading, writing, and searching of information on a database can be better structured and used in various way (such as interoperability functions). In this case the database is queried using python programming language to extract naming descriptions that is user friendly, then codify ID according to the UK national annex recommendation plus all other necessary metadata information. Thereafter, the ID string is returned to a separate table in the database considered as a register for all information created and used. Operation for the creation of container names is controlled using a simple GUI.

Third Tier is the verification of all information that resides in the folders created in Tier 1. The mechanism in this case runs a check across the parent folder of the folder-based CDE to make sure that information is correctly labelled according to the naming format, and to flag-up possible duplications and non-conforming files. The result is displayed on the screen for view check, and it is sorted in two display spaces tagged as conforming files and non-conforming files.

Furthermore, the verification script is designed to check information consistency using the naming convention as stated above. In this case the UK national annex to ISO-19650. The

programme code is written to check the period fields that define container ID according to the recommendation, though it expanded to cover an additional field for description as result of an internal request. The working principles is structured in a way that each field is defined by a certain number of characters it is supposed to hold (for example project = 6 characters, originator = 6 characters, role = 2 etc.) The program checks the naming string of each file to determine if it is true or false, validating it is using the character placeholders. For all that is true they are considered as conforming and displayed in a section for visual examination and likewise for all that is false are considered non-conforming and displayed in a different section.

4.2. Concept Process Flow Diagrams

Figure 24 below is a three-tier illustration of the concept solutions that tackles information management in an Architectural practice during project delivery phase when robust document management system is not used.

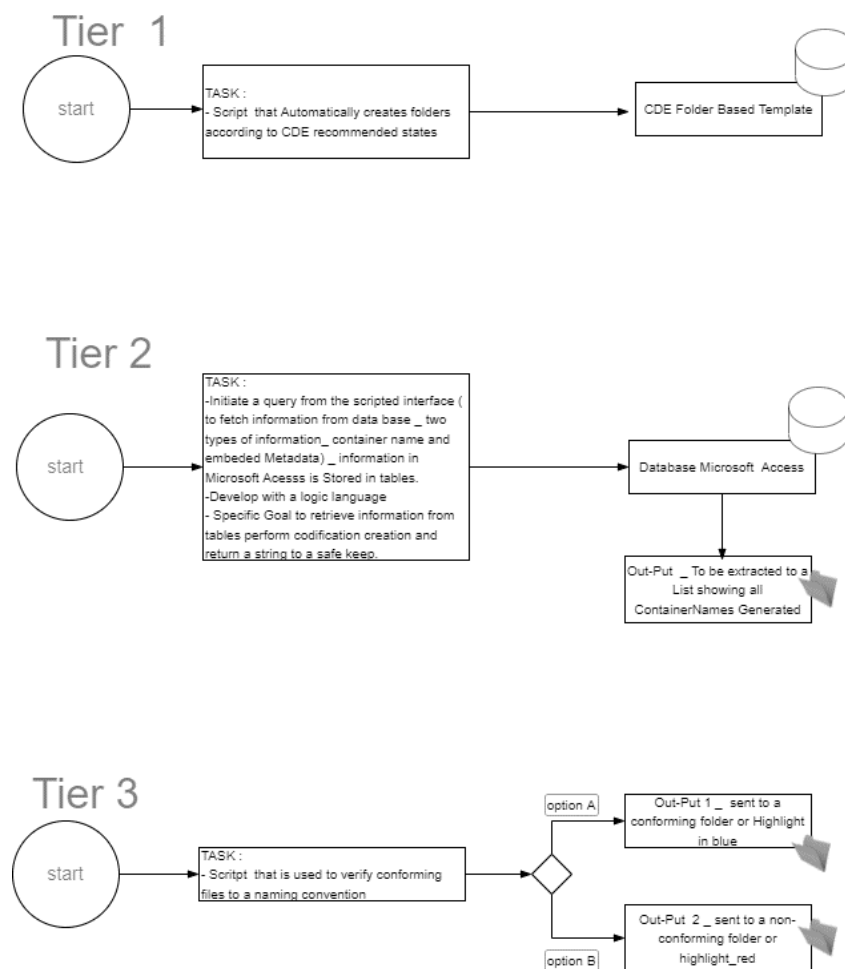


Figure 24 - Diagrammatic representation of the solutions.

4.3. Tier 1: Scripted Folder Based CDE creator

4.3.1. Illustrative Structure

Description: Figure 25 below shows the flowchart of how folders are created automatically. Inputs of naming structure is inserted into various tables according to the folder structure in a database. Aimed at making the creation of folders for any project consistent and accurate. Though, with additional flexibility to expand and modify as per case need. The Python programming language connects to the database using a built-in connector called “pyodbc” (It’s an open-source python module that makes accessing ODBC databases simple). In this case the program fetches information from the respective access tables, which is used to Label each folder. The GUI is an easy-to-use interface that allows the user to make selection of how the folder structure is created (Mainly in a tree like manner).

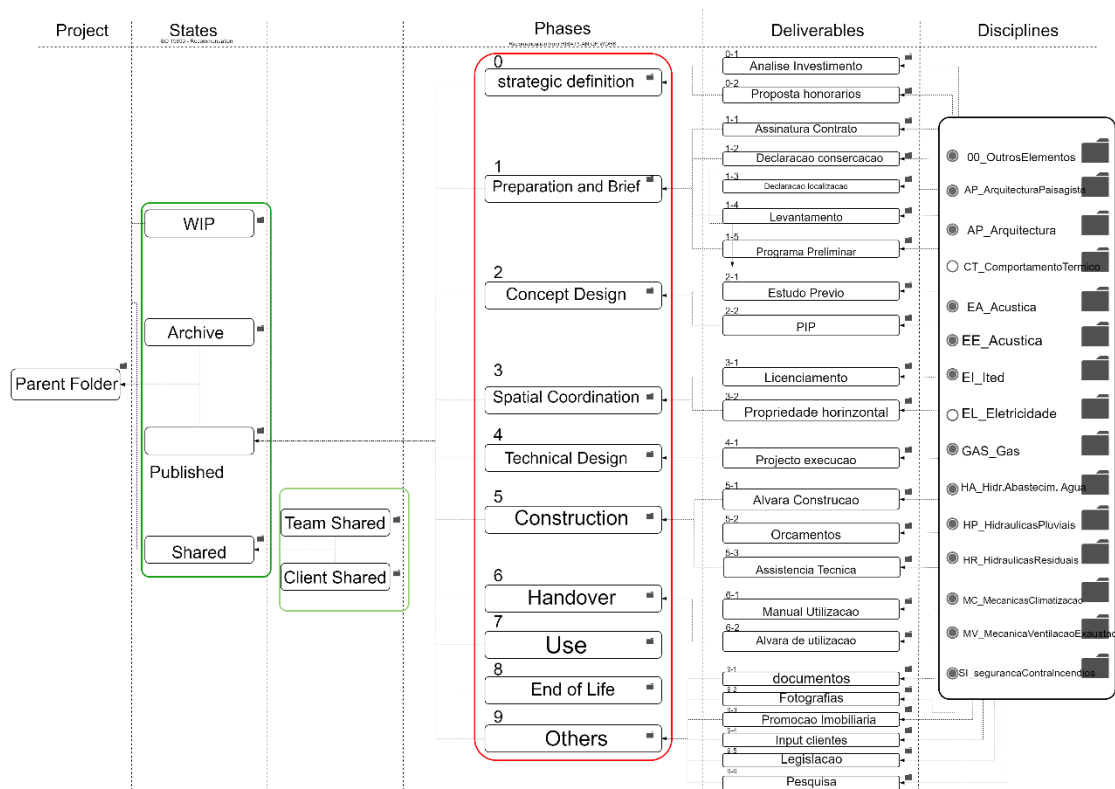


Figure 25 - Illustration of Folder based CDE, according to ISO 19650, which Includes expanded functions prescribed by MC’s practice, Informed by their current practice.

4.3.2. Graphical Interface User Instruction / Workflow Principle

- 1 The package is designed as a standalone programme that can be installed into a Windows operating system. It will need to have Microsoft Access installed as well.
- 2 Upon launching, the displayed interface will have 5 major columns namely: Folder path parent, States, Phases Associated to Published, Deliverables associated to phases, Disciplines associated to deliverables.

Folder path parent: As the name implies is the main holding folder that carries the project Name. it is created first into any desired location within the computer or a server. Every other folder is directed to that location.

States column: Outlines the CDE structure recommend by ISO-19650. WIP, shared, published and Archive respectively, selection for each state is done through a drop list that gets the folder name from the database. Option for additional folders is available.

Sub-section within this column allows for the addition for two partitions into the Shared folder state to House information shared by Teams and Client.

- 3 Column three is an associated column, meaning it is depending on the published state. All Folders selected to be created in this column will be inside the Published state folder. (Consisting of stages 0-9 which corresponds to international standard recommendations).
- 4 Deliverables associated with phases makes the connection between international recommendation and need to have local ground regulations for any practice thereby making it mandatory to group all sections needed as deliverables into each stage of work. (Groups 1 – 9)
- 5 Disciplines associated to deliverables refers to external sharing with collaborators working with the practice. As a rule, only agreed and approved information is to be shared. Hence, all external information to be shared will be in the disciplines folder labelled with the name of the discipline.

Finally, when all selections are made accordingly, a button is clicked to automatically create all folders in their respective placeholder location.

4.4. Tier 2: Scripted container name creator

4.4.1. Illustrative Structure

Description: Figure 26 illustrates the building mechanism for the creation of naming strings. Database organised in Microsoft Access, Logic developed in python language, and the Interface created with “Tkinter” built in python Module. The database in this solution has two purposes, serving as inputs for project admin to insert naming structure and the user returning a string of generated container identification into a table that can be used in future for possible migration to other systems or even interoperable with other common data environment.

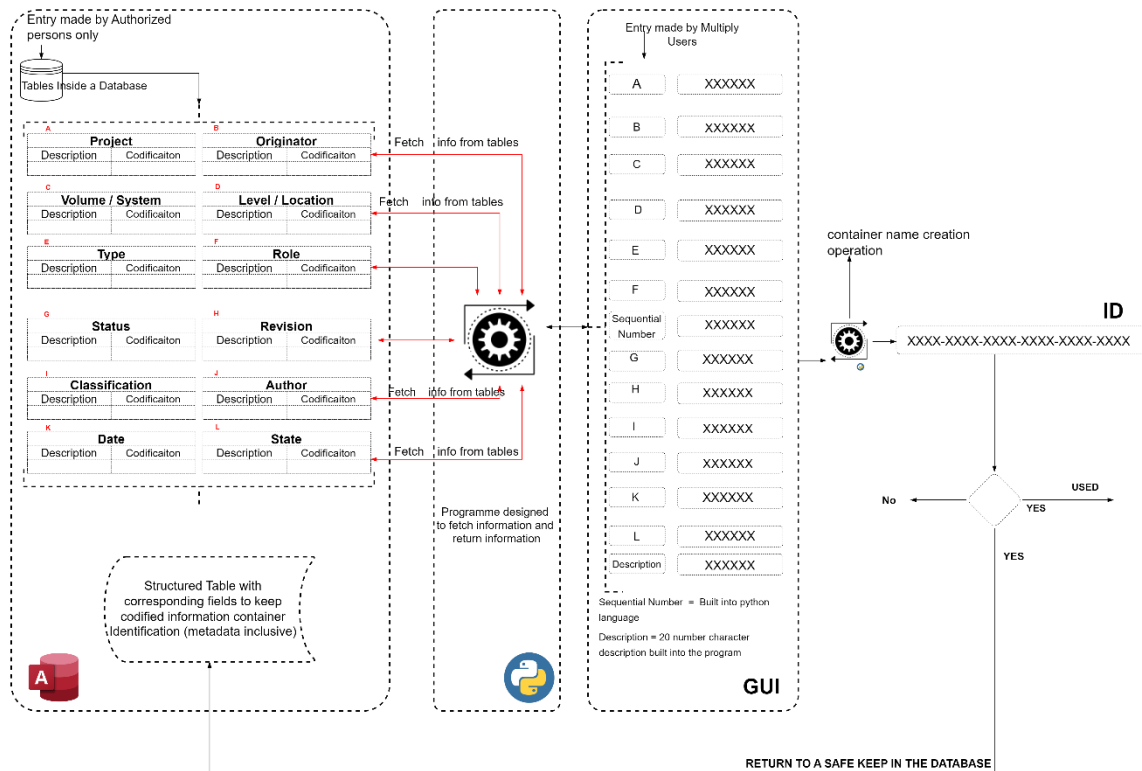


Figure 26 - Illustration of how container names are generated and stored

4.4.2. Graphical Interface User Instruction / Workflow Principle

Step by step instruction on how to use the solution is as follows:

- 1 Launch the GUI as it is a standalone application.

Populate each field with the required information. 14 fields are embedded in table within the database. As project increase so the tables increase. This goes to show that the solution is scalable, and its use is dynamic depending on the project type. The description field is the last field which was incorporated into the solution considering a unique request by MC's practice to enable some form of personalization of each container name. the process that works behind the scenes is one which call out the code associated with the populated description on the interface.

- 2 When the above is settled. On the third row from the top are lists of buttons. From left to right (Add ID, Clear List, Copy ID, Create ID combination, and add to database). Each performs a different function. To display the string which is used for an information container ID the create ID combination should be clicked and automatically that is generated and displayed in the ID field. Once a string is generated three function is designed for its next use. They are copy, addition to a database table and displaying on the interface as a form of temporary tracking
- 3 Hence, Copy ID button is clicked to store out the ID which can be used for to rename information container of any format that is to reside within the CDE folders.

- 4 Further to that, the Add list button is used to add a complete string of all the information concatenated to generate the ID into the Data list at the bottom of the interface. (For visual check)
- 5 Thereafter, it is recommended at intervals of 10 number ID generated. The Add to database button is clicked to return into a table within database.

4.5. Tier 3: Automated verification script

4.5.1. Illustrative Structure

Description: Figure 27 below contrary to the solution 1 and 2 is an automation for the verification of information. It works basically as a check for conforming and non-conforming information within the folder based-CDE. The check is performed based on the naming structure of each container. The rule is designed to sort out into lists correct and wrong files. The verification of information absolutely confirms that all procedures are done correctly.

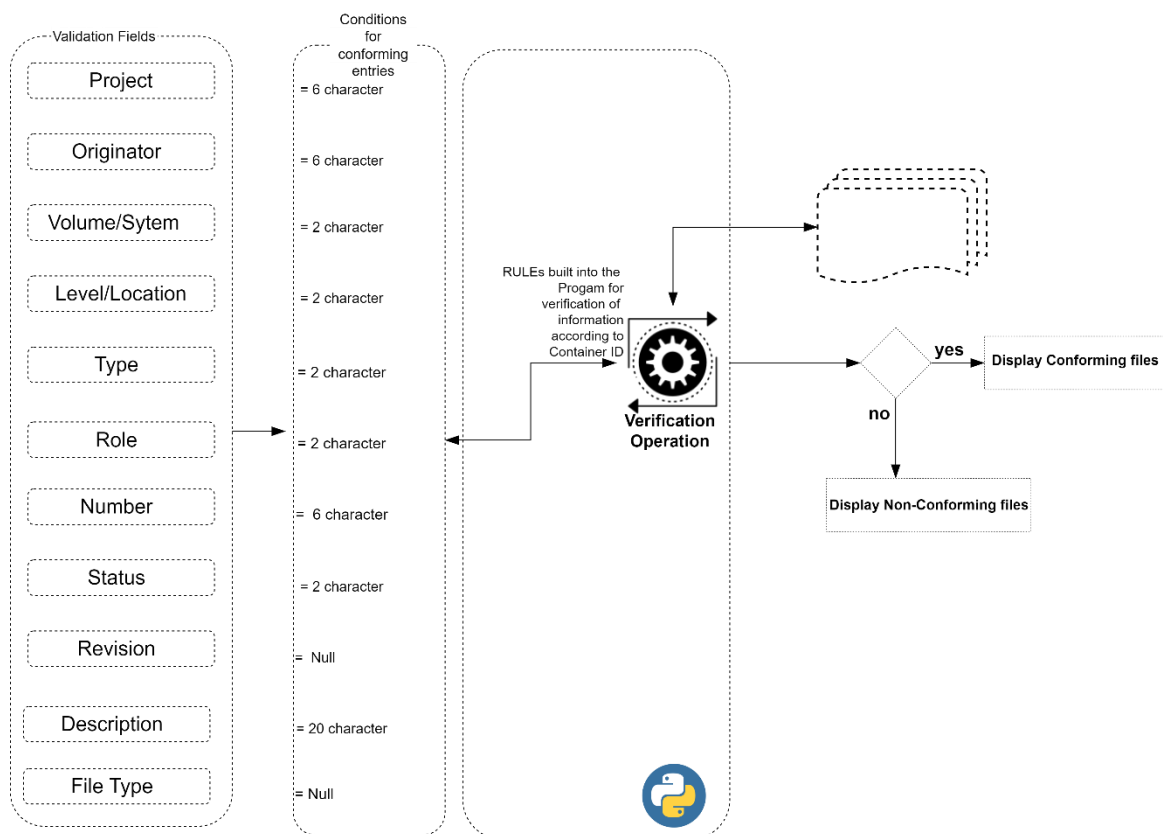


Figure 27 - Illustration of how information is verified

4.5.2. Graphical Interface User Instruction / Workflow Principle

Step by step instruction on how to use the solution is as follows:

- 1 Launch the GUI as it is a standalone application.

- 2 Find the path where the folder to be checked is located
- 3 Activate the check folder items for naming consistency.
- 4 Click on the button (Run Verification) if everything goes as it should be the list box below with sort-out all files into two groups (showing conforming and non-conforming)
- 5 Lastly, a generate report button at the bottom of the interface can be used to export of details of the result.

This page is intentionally left blank

5. CONCLUSION

5.1. General Conclusion

This work revealed that information management using BIM as recommended by ISO-19650 series is key to the advancement of any organisation that adopts it, in this case of study there are positive indicators of improvement of existing information management practices. Although, the notion that large capital investment is required for improved information management using BIM has seen a shift to an alternative approach. Whereby, simple inexpensive solutions can be critically manipulated to achieve or align processes to what much expensive solutions will offer (A glimpse of this possible was found in this study. Making investing capital a less limiting factor).

A part way for small and medium organisations to participate in using recommendation of ISO-19650 series in an affordable manner for managing information was divulged. The picture was to see that a complete BIM TOOL KIT is achieved, but due to time frame allowed for the work and knowledge of soft computer systems development. The work focus to answer folder based CDE start kit.

The Approach was to produce semi-automated 4-layer CDE kit using available resources for information management in a small Architectural practice. Namely, folder based CDE template, container naming creator, scripted verification tool and interoperable mechanism with cloud-base or EDMS common data environments. Three out of four was developed to a significant degree. It improves information management functions in accordance with ISO-19650 recommendation (partially) by automating folder creation (CDE state information placeholder WIP, Shared, Published and Archive states), generating container naming identification strings (for correct and unique labelling of information), and lastly, verification tool that runs a check on all items in the folder for consistency. Furthermore, a database structure was structure to play the role of a registry for safe keep of all information labels with the aim to utilise it for possible interoperability.

Worth to mention that the solutions develop has the potential to reduce fatigue generated from manual processes, improve efficiency, reduce errors, and organise information for easier future use. The solutions developed are generic making them possible to be used by other industry players.

Finally, it is remarkable how much can be achieved with the readily available open-source resources, considering the fact that two-third of global employment is in small and medium scale organisations. The study opens the potential for this significant category of organisational improvement through information management which has a direct influence on business process by indulging in the use of open available and inexpensive solutions to improve information management which in turn will certainly impact market share (business improvement) positively.

5.2. Recommendations

Overall, the studies conducted is a tip of the iceberg as to how much can be achieved through critical investigation for meaningful improvement at the barest cost for establishments with low investment capitals, willing to adopt globally accepted improvement around information management using BIM. However, the outcome of this work has the following recommendations:

- 1 The simple solutions developed in this study opens potential development for further improvement with the aim of developing a complete BIM TOOL KIT for better information management, developed within Architecture practices and with all recommendations of ISO-19650 1&2 aligning with it.
- 2 Based on the possibility coming from the study, which shows that information consistency can be achieve through simple and inexpensive solutions. The big issue around interoperability in Architecture, Engineering, and construction (AEC) can be improved. Through further research on the building frame of electronic management systems which could better give and insight as to how better to structure information that would seamless be mapped across different systems.
- 3 Allied AEC industry professionals that collaborate with Architectural practices can dive into the development of BIM TOOL KIT for better information management and that is cognisance of their unique working processes but at the same time aligns with international standards. Thereby making collaborative operation possible and affordable.

REFERENCES

- Abukhder, J., & Munns, A. K. (2003). Attributing management problems on construction projects to project information. In *Association of Researchers in Construction Management* (Vol. 2).
- Agarwal R., Chandrasekaran S., & Sridhar M. (2020, October 10). *Imagining construction's digital future | McKinsey*. Imagining Construction's Digital Future. <https://www.mckinsey.com/business-functions/operations/our-insights/imagining-constructions-digital-future>
- Bourgeois, D., & Bourgeois, D. T. (2014). *Chapter 1: What Is an Information System?* Published through the Open Textbook Challenge by the Saylor Academy.
- Brennen, J. S., & Kreiss, D. (2016). Digitalization. In *The International Encyclopedia of Communication Theory and Philosophy* (pp. 1–11). Wiley. <https://doi.org/10.1002/9781118766804.wbiect111>
- Cornick, T. C., & Barre, N. J. (1991). Quality Management and Design-Build: The Opportunities for this Method of Procurement. *International Journal of Quality & Reliability Management*, 8(3). <https://doi.org/10.1108/02656719110135076>
- Dawood, N., Akinsola, A., & Hobbs, B. (2002). *Development of automated communication of system for managing site information using internet technology*.
- EN ISO 19650-1. (2018). *ISO_19650-1_2018(en)*.
- Marta Campos Atelier de Arquitectura. (n.d.). *About - Marta Campos*. Retrieved August 20, 2022, from <http://martacampos.pt/en/about/>
- Mirus, P. (2017). (3) *Information Management Basics - YouTube*. <https://www.youtube.com/watch?v=2pMl0bHxEck&t=1557s>
- RIBA. (2020). *RIBA Plan of Work 2020 Overview RIBA Plan of Work*. www.ribaplanofwork.com
- Saarikko, T., Westergren, U. H., & Blomquist, T. (2020). Digital transformation: Five recommendations for the digitally conscious firm. *Business Horizons*, 63(6), 825–839. <https://doi.org/10.1016/j.bushor.2020.07.005>
- shillcock, P. (2022). (3) *ISO 19650 Overivew: The Who, Why, What, and How of ISO 19650 by Paul Shillcock (Operam) - YouTube*. <https://www.youtube.com/watch?v=l-P0X7gXwNg>
- Succar, B. (2010). *Building Information Modelling Maturity Matrix*.

- Techopedia. (2017). *What is an Information Management System (IMS)? - Definition from Techopedia.* <https://www.techopedia.com/definition/26859/information-management-system>
- Turk, Ž. (2020). Interoperability in construction – Mission impossible? *Developments in the Built Environment*, 4. <https://doi.org/10.1016/j.dibe.2020.100018>
- UK BIM Framework. (2021). *ISO 19650 Guidance Part A: The information management function and resources.* <https://ukbimframeworkguidance.notion.site/ISO-19650-Guidance-Part-A-The-information-management-function-and-resources-cf1be61f9d2345e59abe07a4d740e751>
- Uk BIM Framework. (2021). *ISO 19650 Guidance Part C: Facilitating the CDE (workflow and technical solutions).* <https://ukbimframeworkguidance.notion.site/ISO-19650-Guidance-Part-C-Facilitating-the-CDE-workflow-and-technical-solutions-ff3bdbcf1c1349c1a98c586943d0a9f1>
- van de Watt, W. (2022). (3) *1tag part 2 - information Management Workflow [03-P0101-S0] - YouTube.* Information Management Workflow. https://www.youtube.com/watch?v=GvGmAwrM0_M
- Werner, R. (2020). (3) *The Father Of Quantitative Easing - Richard Werner | PBD Podcast | Ep. 161 - YouTube.* https://www.youtube.com/watch?v=kDmY1kY4slk&list=PLQ-rUC37UWfNNU4XtJhMfNzfCTaytD_RV&index=18&t=1722s

LIST OF ACRONYMS AND ABBREVIATIONS

AEC	Architecture, Engineering and Construction
AECO	Architecture, Engineering, Construction and Operation
BIM	Building Information Modelling
CDE	Common data Environment
DBMS	Data-base Management system
DDD	Digital Transformation, Digitalization, Digitization
EDMS	Electronic Document management system
GUI	Graphic User Interface
ISO	International Standard organisation
IMS	Information management systems
ID	Identification
MC	Marta Campos Atelier de Arquitectura
RIBA	Royal Institute of British Architects
TTT	Triple T solutions
WIP	Work in Progress

APPENDICES

Appendix 1: Code used for solution 1

```

1  import tkinter
2  import os
3  from tkinter.filedialog import askdirectory
4  import tkinter.messagebox
5  import tkinter as tk
6  from tkinter import END, N, RIDGE, StringVar, ttk
7  from turtle import fillcolor
8  from typing import Text
9  from xml.dom.pulldom import END_DOCUMENT
10
11 from setuptools import Command
12
13 #=====#
14
15 import pyodbc
16 conn = pyodbc.connect(
17     r'Driver={Microsoft Access Driver (*.mdb,
18         *.accdb)};DBQ=C:/Users/KNIGHT_PC/Downloads/Thesis-dev/ID.accdb;')
19 # conn = pyodbc.connect(
20 #     r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:/Sch_Wrk/DB.accdb;')
21 cursor = conn.cursor()
22
23 sql1 = "SELECT Folders FROM States_tbl"
24 cursor.execute(sql1)
25 teste_shared = ["TEAM_SHARED", "CLIENT_SHARED"]
26 teste = []
27
28 for row in cursor.fetchall():
29     teste.append(row.Folders)
30
31 sql2 = "SELECT Folders FROM Phases_tbl"
32 cursor.execute(sql2)
33 # ID = cursor.fetchall()
34 testel = []
35
36 for row in cursor.fetchall():
37     testel.append(row.Folders)
38
39 sql4 = "SELECT Folders FROM Deliverables_tbl"
40 cursor.execute(sql4)
41 deliver = []
42 for row in cursor.fetchall():
43     deliver.append(row.Folders)
44
45 sql5 = "SELECT Folder FROM Disciplines_tbl"
46 cursor.execute(sql5)
47 disciplines = []
48 for row in cursor.fetchall():
49     disciplines.append(row.Folder)
50 #===== frames
51 #=====#
52
53 # Dir select
54
55 def selectPath():
56     path_ = askdirectory()
57     path.set(path_)
58
59
60 def create_folder():
61     dirs = os.path.join(path.get(), folder.get())
62     if not os.path.exists(dirs):
63         # To save the main path of the parent folder
64         global main_path

```

```

65     main_path = path.get() + "/" + folder.get()
66     os.makedirs(dirs)
67     tkinter.messagebox.showinfo(
68         'Tips:', 'Folder name created successfully!')
69 else:
70     tkinter.messagebox.showerror(
71         'Tips:', 'The folder name exists, please change it')
72
73
74 def create_folders():
75     # If Parent Folder has not been created, throw an error
76     try:
77         if main_path == "":
78             tkinter.messagebox.showerror('Tips', 'Please Create Parent Folder')
79         use_pass = use_var.get()
80         end_pass = end_var.get()
81         SD_deliverables = [deliver0.get(), deliver1.get()]
82         PB_deliverables = [deliver2.get(), deliver3.get(),
83                             deliver4.get(), deliver5.get(), deliver6.get()]
84         CD_deliverables = [deliver7.get(), deliver8.get()]
85         SC_deliverables = [deliver9.get(), deliver10.get()]
86         TD_deliverables = [deliver11.get()]
87         C_deliverables = [deliver12.get(), deliver13.get()]
88         H_deliverables = [deliver14.get(), deliver15.get()]
89         U_deliverables = [use_pass]
90         E_deliverables = [end_pass]
91         O_deliverables = [deliver16.get(), deliver17.get(),
92                             deliver18.get(), deliver19.get(), deliver20.get(), deliver21.
93                             get()]
94     global raw, sub_shared, states, publish_path, shared_path, array_of_deliverables,
95     deliverables, disciplines
96     deliverables = []
97     disciplines_array = [discipline_0, discipline_1, discipline_2, discipline_3,
98                         discipline_4, discipline_5, discipline_6, discipline_7,
99                         discipline_8, discipline_9, discipline_10, discipline_11,
100                        discipline_12, discipline_13, discipline_14, discipline_15]
101     array_of_deliverables = [SD_deliverables, PB_deliverables, CD_deliverables,
102                             SC_deliverables,
103                             TD_deliverables, C_deliverables, H_deliverables,
104                             U_deliverables, E_deliverables, O_deliverables]
105
106     # Defining Paths
107     publish_path = main_path + "/" + "PUBLISHED"
108     shared_path = main_path + "/" + "SHARED"
109     raw = [state1, state2, state3, state4, state5, state6, state7, state8]
110     phases = [phase0, phase1, phase3, phase4, phase5, phase6,
111               phase7, phase8, phase9, phase10, phase11, phase12]
112     sub_shared, states = [], []
113     # Establish where Folders Should be Stored
114     for i in raw:
115         current_value = i.get()
116         if current_value != "" and current_value != "TEAM_SHARED" and current_value
117         != "CLIENT_SHARED":
118             states.append(current_value)
119         if current_value == "TEAM_SHARED" or current_value == "CLIENT_SHARED":
120             sub_shared.append(current_value)
121     # Create Folders under Parent Folder
122     for x in states:
123         create_sub_dir(main_path, x)
124     # Create Folders under SHARED folder
125     for y in sub_shared:
126         create_sub_dir(shared_path, y)
127     # Create Folders under Published
128     for p in phases:
129         current_value = p.get()
130         if current_value != "":
131             create_sub_dir(publish_path, current_value)
132

```

```

125         # Create Deliverables
126         for i in range(len(testel)):
127             path = publish_path + "/" + testel[i]
128             if i < 10:
129                 for j in array_of_deliverables[i]:
130                     added_path = j
131                     if added_path != "":
132                         d_path = path + "/" + added_path
133                         deliverables.append(d_path)
134                         create_sub_dir(path, j)
135
136         # Create Disciplines
137         for d in deliverables:
138             for i in disciplines_array:
139                 current_value = i.get()
140                 if current_value != "":
141                     create_sub_dir(d, current_value)
142             tkinter.messagebox.showinfo(
143                 'Tips:', 'Folders created successfully!')
144     except:
145         tkinter.messagebox.showerror(
146             'Tips:', 'Something Went Wrong')
147
148     # for y in deliverables:
149
150
151 def create_sub_dir(send_path, state):
152     path = os.path.join(send_path, state)
153     os.makedirs(path)
154     # create clear function
155     # def clear():
156         #txt_box.delete(1.0, END)
157
158     # grab text
159     # def Get_Text():
160         # pass
161
162
163     #=====
164     if __name__ == '__main__':
165         root = tkinter.Tk()
166         root.title("cde folder creator")
167         # root.geometry("1600x768")
168         # root.resizable(width=False, height=False)
169         # root.iconbitmap("C:/Sch_Wrk/logo.ico")
170         root.iconbitmap(
171             "C:/Users/KNIGHT_PC/Downloads/Thesis-dev/logo.ico")
172
173     #=====
174
175     main_frame = tkinter.Frame(root)
176     main_frame.pack()
177     #=====
178
179     # folder path selection
180     path = tkinter.StringVar()
181     # folder name selection
182     folder = tkinter.StringVar()
183     # wip folder
184     wip = tkinter.StringVar()
185     wip = "WIP"
186     # shared folder
187     shared = tkinter.StringVar()
188     shared = "SHARED"

```

```

189     # published folder
190     published = tkinter.StringVar()
191     published = "PUBLISHED"
192     # archive folder
193     archive = tkinter.StringVar()
194     archive = "ARCHIVE"
195
196     #=====
197
198
199     #===== Main folder path
200     #=====
201
202     # location path & parent folder location
203     prt_ltn_frame = tkinter.LabelFrame(
204         main_frame, text=" Folder Path Parent ", relief=RIDGE)
205     prt_ltn_frame.grid(row=0, column=0, sticky="ns", padx=3, pady=3)
206
207     # create path label_entry_button
208     prt_ltn_label = tkinter.Label(
209         prt_ltn_frame, text="Target Path:", fg="grey").grid(row=0, column=0, sticky="w")
210
211     prt_ltn_label = tkinter.Entry(prt_ltn_frame, textvariable=path).grid(
212         row=1, column=0, padx=10, pady=10)
213
214     prt_ltn_button = tkinter.Button(prt_ltn_frame, text="Select Path:", command=
215         selectPath).grid(
216         row=2, column=0, padx=10, pady=10, sticky="news")
217
218     #=====
219
220     # create separator btw Target & folder name
221     #sep = ttk.Separator(prt_ltn_frame, orient="vertical")
222     #sep.grid(row=0, column=3, sticky="NS")
223
224     #=====
225
226     # create folder label_entry_button
227     prt_folder_label = tkinter.Label(
228         prt_ltn_frame, text="Folder Name", fg="grey").grid(row=3, column=0, sticky="w")
229
230     prt_folder_label = tkinter.Entry(prt_ltn_frame, textvariable=folder).grid(
231         row=4, column=0, padx=20, pady=10)
232
233     prt_folder_button = tkinter.Button(prt_ltn_frame, text="Submit", command=
234         create_folder).grid(row=5, column=0,
235         padx=20, pady=10, sticky="news")
236
237     for widget in prt_ltn_frame.winfo_children():
238         widget.grid_configure(padx=5, pady=5)
239
240     #=====states=====
241     #=====
242
243     # folder state wideget & db connection
244     states_frame = tkinter.LabelFrame(
245         main_frame, text=" States ", relief=RIDGE)
246     states_frame.grid(row=0, column=1, sticky="ns", padx=3, pady=3)
247
248     # create states folder

```

```

248     global statel
249     statel = tk.StringVar()
250     wip_state = tkinter.Label(
251         states_frame, text="WIP", fg="grey").grid(row=0, column=0, sticky="w")
252     wip_state = ttk.Combobox(states_frame, values=wip,
253                             textvariable=statel).grid(row=0, column=1)
254
255     #self.prj = ttk.Combobox(rcrds_Label, values = self.ID, width=30).grid(row=0,
256     column=1, padx=5, pady=5)
257     global state2
258     state2 = tk.StringVar()
259     Shared_state = tkinter.Label(
260         states_frame, text="SHARED", fg="grey").grid(row=1, column=0)
261     Shared_state = ttk.Combobox(
262         states_frame, values=shared, textvariable=state2).grid(row=1, column=1)
263
264     global state3
265     state3 = tk.StringVar()
266     Published_state = tkinter.Label(
267         states_frame, text="PUBLISHED", fg="grey").grid(row=2, column=0)
268     Published_state = ttk.Combobox(
269         states_frame, values=published, textvariable=state3).grid(row=2, column=1)
270
271     global state4
272     state4 = tk.StringVar()
273     Archive_state = tkinter.Label(
274         states_frame, text="ARCHIVED", fg="grey").grid(row=3, column=0)
275     Archive_state = ttk.Combobox(
276         states_frame, values=archive, textvariable=state4).grid(row=3, column=1)
277
278     global state5
279     state5 = tk.StringVar()
280     Custom_1 = tkinter.Label(states_frame, text="CUSTOM1",
281                             fg="grey").grid(row=4, column=0)
282     Custom_1 = ttk.Combobox(states_frame, values=teste,
283                             textvariable=state5).grid(row=4, column=1)
284
285     global state6
286     state6 = tk.StringVar()
287     Custom_2 = tkinter.Label(states_frame, text="CUSTOM2",
288                             fg="grey").grid(row=5, column=0)
289     Custom_2 = ttk.Combobox(states_frame, values=teste,
290                             textvariable=state6).grid(row=5, column=1)
291
292     # create separator btw state folder and subfolder
293     sep1 = ttk.Separator(states_frame, orient="horizontal")
294     sep1.grid(row=6, column=0, columnspan=2, sticky="NS")
295
296     subfolder_shared = tkinter.Label(
297         states_frame, text="SHARED Subfolder:").grid(row=7, column=0)
298
299     global state7, state8
300     state7, state8 = tk.StringVar(), tk.StringVar()
301     team_Shared = tkinter.Label(
302         states_frame, text="Team_Shared", fg="grey").grid(row=8, column=0)
303     team_Shared = ttk.Combobox(
304         states_frame, values=teste_shared[0], textvariable=state7).grid(row=8, column=1)
305
306     client_shared = tkinter.Label(
307         states_frame, text="Client_Shared", fg="grey").grid(row=9, column=0)
308     client_shared = ttk.Combobox(
309         states_frame, values=teste_shared[1], textvariable=state8).grid(row=9, column=1)
310
311     for widget in states_frame.winfo_children():
312         widget.grid_configure(padx=2, pady=3, sticky="w")
313

```

```

314 #=====phases=====
315 # folder Phases
316 phases_frame = tkinter.LabelFrame(
317     main_frame, text="_Phases Associated to Published", relief=RIDGE)
318 phases_frame.grid(row=0, column=2, sticky="news", padx=3, pady=3)
319
320 ##text= {"0_Str/1_Pre/2_Conc/3_Spat/4_Tech/5_Co/6_Han/7_Use/8_End of Life/9_Others"}
321 # , fg="grey").grid(row=0, column=0)
322
323 # create Phases
324 global phase0, phase1, phase2
325 phase0, phase1, phase2 = tk.StringVar(), tk.StringVar(), tk.StringVar()
326 str_def = tkinter.Label(phases_frame, text="0_",
327     fg="grey").grid(row=0, column=0)
328 str_def = ttk.Combobox(phases_frame, values=testel[0], textvariable=phase0).grid(
329     row=0, column=1, sticky="w")
330
331 prep_brief = tkinter.Label(
332     phases_frame, text="1_", fg="grey").grid(row=1, column=0)
333 prep_brief = ttk.Combobox(phases_frame, values=testel[1], textvariable=phase1).grid(
334     row=1, column=1, sticky="w")
335
336 con_dsgn = tkinter.Label(phases_frame, text="2_",
337     fg="grey").grid(row=2, column=0)
338 con_dsgn = tkinter.ttk.Combobox(
339     phases_frame, values=testel[2], textvariable=phase2).grid(row=2, column=1, sticky
340     ="w")
341
342 global phase3, phase4, phase5
343 phase3, phase4, phase5 = tk.StringVar(), tk.StringVar(), tk.StringVar()
344 spt_coor = tkinter.Label(phases_frame, text="3_",
345     fg="grey").grid(row=3, column=0)
346 spt_coor = tkinter.ttk.Combobox(
347     phases_frame, values=testel[3], textvariable=phase3).grid(row=3, column=1, sticky
348     ="w")
349
350 tech_dsgn = tkinter.Label(
351     phases_frame, text="4_", fg="grey").grid(row=4, column=0)
352 tech_dsgn = tkinter.ttk.Combobox(
353     phases_frame, values=testel[4], textvariable=phase4).grid(row=4, column=1, sticky
354     ="w")
355
356 # create separator btw state folder and subfolder
357 #sep2 = ttk.Separator(phases_frame, orient="horizontal")
358 #sep2.grid(row=1, columnspan=5, sticky="EW")
359
360 global phase6, phase7, phase8
361 phase6, phase7, phase8 = tk.StringVar(), tk.StringVar(), tk.StringVar()
362 construc = tkinter.Label(
363     phases_frame, text="5_", fg="grey").grid(row=5, column=0)
364 construc = tkinter.ttk.Combobox(
365     phases_frame, values=testel[5], textvariable=phase5).grid(row=5, column=1, sticky
366     ="w")
367
368 handover = tkinter.Label(phases_frame, text="6_",
369     fg="grey").grid(row=6, column=0)
370 handover = tkinter.ttk.Combobox(
371     phases_frame, values=testel[6], textvariable=phase6).grid(row=6, column=1, sticky
372     ="w")
373
374 use = tkinter.Label(phases_frame, text="7_",
375     fg="grey").grid(row=7, column=0)
376 use = tkinter.ttk.Combobox(phases_frame, values=testel[7], textvariable=phase7).grid(
377     row=7, column=1, sticky="w")

```

```

375     global phase9, phase10, phase11, phase12
376     phase9, phase10, phase11, phase12 = tk.StringVar(
377     ), tk.StringVar(), tk.StringVar(), tk.StringVar()
378     end_of_life = tkinter.Label(
379     phases_frame, text="8_", fg="grey").grid(row=8, column=0)
380     end_of_life = tkinter.ttk.Combobox(
381     phases_frame, values=testel[8], textvariable=phase8).grid(row=8, column=1, sticky
382     ="w")
383
384     others = tkinter.Label(phases_frame, text="9_",
385     fg="grey").grid(row=9, column=0)
386     others = tkinter.ttk.Combobox(phases_frame, values=testel[9], textvariable=phase9).
387     grid(
388     row=9, column=1, sticky="w")
389
390     Custom_1 = tkinter.Label(phases_frame, text="Ctmz_1",
391     fg="grey").grid(row=10, column=0)
392     Custom_1 = ttk.Combobox(phases_frame, values=teste,
393     textvariable=phase10).grid(row=10, column=1)
394
395     Custom_2 = tkinter.Label(phases_frame, text="Ctmz_2",
396     fg="grey").grid(row=11, column=0)
397     Custom_2 = ttk.Combobox(phases_frame, values=teste,
398     textvariable=phase11).grid(row=11, column=1)
399
400     Custom_3 = tkinter.Label(phases_frame, text="Ctmz_3",
401     fg="grey").grid(row=12, column=0)
402     Custom_3 = ttk.Combobox(phases_frame, values=teste,
403     textvariable=phase12).grid(row=12, column=1)
404
405     for widget in phases_frame.winfo_children():
406         widget.grid_configure(padx=5, pady=5)
407
408     #=====deliverables=====
409     #
410     # folder deliverables
411     deliverables_frame = tkinter.LabelFrame(
412     main_frame, text=" Deliverables associated to Phases_", relief=RIDGE)
413     deliverables_frame.grid(row=0, column=3, sticky="news", padx=1, pady=1)
414
415     # create separator
416     #sep3 = ttk.Separator(deliverables_frame, orient="horizontal")
417     #sep3.grid(row=0, columnspan=6, sticky="EW")
418
419     global deliver0, deliver1
420     deliver0, deliver1 = tk.StringVar(), tk.StringVar()
421     str_def = tkinter.Label(deliverables_frame, text="0_1", fg="grey").grid(
422     row=0, column=0, sticky="w")
423
424     Análise_Invest = ttk.Combobox(
425     deliverables_frame, values=deliver[0], textvariable=deliver0).grid(row=0, column=
426     1)
427
428     str_def1 = tkinter.Label(deliverables_frame, text="0_2", fg="grey").grid(
429     row=1, column=0, sticky="w")
430     Prop_Hono = ttk.Combobox(
431     deliverables_frame, values=deliver[1], textvariable=deliver1).grid(row=1, column=
432     1)
433
434     #=====
435     #
436
437     sep4 = ttk.Separator(deliverables_frame, orient="horizontal")
438     sep4.grid(row=2, columnspan=2, sticky="ew")

```

```

#=====
=====#1
435
436 global deliver2, deliver3, deliver4, deliver5, deliver6
437 deliver2, deliver3, deliver4, deliver5, deliver6 = tk.StringVar(
438 ), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar()
439
440 Prep_brief = tkinter.Label(deliverables_frame, text="1_1", fg="grey").grid(
441     row=3, column=0, sticky="w")
442 Ass_Cont = ttk.Combobox(deliverables_frame, values=deliver[2], textvariable=deliver2
443 ).grid(
444     row=3, column=1, sticky="w")
445
446 Prep_brief1 = tkinter.Label(deliverables_frame, text="1_2", fg="grey").grid(
447     row=4, column=0, sticky="w")
448 Dec_Cont = ttk.Combobox(deliverables_frame, values=deliver[3], textvariable=deliver3
449 ).grid(
450     row=4, column=1, sticky="w")
451
452 Prep_brief2 = tkinter.Label(deliverables_frame, text="1_3", fg="grey").grid(
453     row=5, column=0, sticky="w")
454 Decl_loc = ttk.Combobox(deliverables_frame, values=deliver[4], textvariable=deliver4
455 ).grid(
456     row=5, column=1, sticky="w")
457
458 Prep_brief3 = tkinter.Label(deliverables_frame, text="1_4", fg="grey").grid(
459     row=6, column=0, sticky="w")
460 Levan = ttk.Combobox(deliverables_frame, values=deliver[5], textvariable=deliver5).
461 grid(
462     row=6, column=1, sticky="w")
463
464 Prep_brief4 = tkinter.Label(deliverables_frame, text="1_5", fg="grey").grid(
465     row=7, column=0, sticky="w")
466 Prog_prel = ttk.Combobox(deliverables_frame, values=deliver[6], textvariable=deliver6
467 ).grid(
468     row=7, column=1, sticky="w")
469
470 #=====
471 #
472 sep5 = ttk.Separator(deliverables_frame, orient="horizontal")
473 sep5.grid(row=8, columnspan=2, sticky="ew")
474
475 #=====
476 #2
477 global deliver7, deliver8
478 deliver7, deliver8 = tk.StringVar(), tk.StringVar()
479
480 concept_design = tkinter.Label(
481     deliverables_frame, text="2_1", fg="grey").grid(row=9, column=0, sticky="w")
482 Estudo_Prévio = ttk.Combobox(deliverables_frame, values=deliver[7], textvariable=
483 deliver7).grid(
484     row=9, column=1, sticky="w")
485
486 concept_design1 = tkinter.Label(
487     deliverables_frame, text="2_1", fg="grey").grid(row=10, column=0, sticky="w")
488 pip = ttk.Combobox(deliverables_frame, values=deliver[8], textvariable=deliver8).grid
489 (
490     row=10, column=1, sticky="w")
491
492 #=====
493 #
494 sep6 = ttk.Separator(deliverables_frame, orient="horizontal")
495 sep6.grid(row=11, columnspan=2, sticky="ew")
496
497 #=====
498 #3
499
500 global deliver9, deliver10
501 deliver9, deliver10 = tk.StringVar(), tk.StringVar()
502

```

```

489     spl_Crdtn = tkinter.Label(
490         deliverables_frame, text="3_1", fg="grey").grid(row=12, column=0)
491     licenciamento = ttk.Combobox(deliverables_frame, values=deliver[9], textvariable=
492         deliver9).grid(
493         row=12, column=1, sticky="W")
494     spl_Crdtn1 = tkinter.Label(
495         deliverables_frame, text="3_1", fg="grey").grid(row=13, column=0)
496     propriedade_horiz = ttk.Combobox(
497         deliverables_frame, values=deliver[10], textvariable=deliver10).grid(row=13,
498         column=1, sticky="W")
499     #=====
500     #
501     sep7 = ttk.Separator(deliverables_frame, orient="horizontal")
502     sep7.grid(row=14, columnspan=2, sticky="ew")
503     #=====
504     #4
505     global deliver11
506     deliver11 = tk.StringVar()
507     tec_dsgn = tkinter.Label(deliverables_frame, text="4_1", fg="grey").grid(
508         row=15, column=0, sticky="W")
509     tec_dsgn = ttk.Combobox(deliverables_frame, values=deliver[11], textvariable=
510         deliver11).grid(
511         row=15, column=1, sticky="W")
512     #=====
513     #
514     # create separator
515     sep8 = ttk.Separator(deliverables_frame, orient="horizontal")
516     sep8.grid(row=16, columnspan=2, sticky="ew")
517     #=====
518     #5
519     global deliver12, deliver13, deliver25
520     deliver12, deliver13, deliver25 = tk.StringVar(), tk.StringVar(), tk.StringVar()
521     construc = tkinter.Label(deliverables_frame, text="5_1", fg="grey").grid(
522         row=17, column=0, sticky="W")
523     Alv_Const = ttk.Combobox(deliverables_frame, values=deliver[25], textvariable=
524         deliver25).grid(
525         row=17, column=1, sticky="W")
526     construc = tkinter.Label(deliverables_frame, text="5_2", fg="grey").grid(
527         row=18, column=0, sticky="W")
528     orçamentos = ttk.Combobox(deliverables_frame, values=deliver[12], textvariable=
529         deliver12).grid(
530         row=18, column=1, sticky="W")
531     construc = tkinter.Label(deliverables_frame, text="5_3", fg="grey").grid(
532         row=19, column=0, sticky="W")
533     Assi_Téc = ttk.Combobox(deliverables_frame, values=deliver[13], textvariable=
534         deliver13).grid(
535         row=19, column=1, sticky="W")
536     #=====
537     #
538     # create separator
539     sep9 = ttk.Separator(deliverables_frame, orient="vertical")
540     sep9.grid(row=0, column=2, rowspan=20, sticky="ns")
541     #=====
542     #5
543     global deliver14, deliver15

```

```

542 deliver14, deliver15 = tk.StringVar(), tk.StringVar()
543 handover = tkinter.Label(deliverables_frame, text="6_1", fg="grey").grid(
544     row=0, column=3, sticky="W")
545 mum = ttk.Combobox(deliverables_frame, values=deliver[14], textvariable=deliver14).
546     grid(
547         row=0, column=4, sticky="W")
548 handover1 = tkinter.Label(deliverables_frame, text="6_2", fg="grey").grid(
549     row=1, column=3, sticky="W")
550 Alv_de_utl = ttk.Combobox(deliverables_frame, values=deliver[15], textvariable=
551     deliver15).grid(
552     row=1, column=4, sticky="W")
553
554 #=====
555 # create separator
556 sep10 = ttk.Separator(deliverables_frame, orient="horizontal")
557 sep10.grid(row=2, column=3, columnspan=2, sticky="ew")
558
559 #=====
560 #5
561 global use_var
562 use_var = tkinter.StringVar()
563 use = tkinter.Label(deliverables_frame, text="7_1",
564     fg="grey").grid(row=3, column=3, sticky="W")
565 blank = ttk.Combobox(deliverables_frame, textvariable=use_var).grid(
566     row=3, column=4, sticky="W")
567
568 #=====
569 # create separator
570 sep11 = ttk.Separator(deliverables_frame, orient="horizontal")
571 sep11.grid(row=4, column=3, columnspan=2, sticky="ew")
572
573 #=====
574 #5
575 global end_var
576 end_var = tkinter.StringVar()
577 end = tkinter.Label(deliverables_frame, text="8_1",
578     fg="grey").grid(row=5, column=3, sticky="W")
579 end = ttk.Combobox(deliverables_frame, textvariable=end_var).grid(
580     row=5, column=4, sticky="W")
581
582 #=====
583 # create separator
584 sep11 = ttk.Separator(deliverables_frame, orient="horizontal")
585 sep11.grid(row=6, column=3, columnspan=2, sticky="ew")
586
587 #=====
588 #5
589 global deliver16, deliver17, deliver18, deliver19, deliver20, deliver21
590 deliver16, deliver17, deliver18, = tk.StringVar(), tk.StringVar(), tk.StringVar()
591 deliver19, deliver20, deliver21 = tk.StringVar(), tk.StringVar(), tk.StringVar()
592
593 others = tkinter.Label(deliverables_frame, text="9_1", fg="grey").grid(
594     row=7, column=3, sticky="W")
595 documentos = ttk.Combobox(deliverables_frame, values=deliver[16], textvariable=
596     deliver16).grid(
597     row=7, column=4, sticky="W")
598
599 others1 = tkinter.Label(deliverables_frame, text="9_2", fg="grey").grid(
600     row=8, column=3, sticky="W")

```

```

594     fotografias = ttk.Combobox(deliverables_frame, values=deliver[17], textvariable=
deliver17).grid(
595         row=8, column=4, sticky="W")
596
597     others2 = tkinter.Label(deliverables_frame, text="9_3", fg="grey").grid(
598         row=9, column=3, sticky="W")
599     prom_imobi = ttk.Combobox(deliverables_frame, values=deliver[18], textvariable=
deliver18).grid(
600         row=9, column=4, sticky="W")
601
602     others3 = tkinter.Label(deliverables_frame, text="9_4", fg="grey").grid(
603         row=10, column=3, sticky="W")
604     input_cliente = ttk.Combobox(deliverables_frame, values=deliver[19], textvariable=
deliver19).grid(
605         row=10, column=4, sticky="W")
606
607     others4 = tkinter.Label(deliverables_frame, text="9_5", fg="grey").grid(
608         row=11, column=3, sticky="W")
609     legislação = ttk.Combobox(deliverables_frame, values=deliver[20], textvariable=
deliver20).grid(
610         row=11, column=4, sticky="W")
611
612     others5 = tkinter.Label(deliverables_frame, text="9_6", fg="grey").grid(
613         row=12, column=3, sticky="W")
614     pesquisa = ttk.Combobox(deliverables_frame, values=deliver[21], textvariable=
deliver21).grid(
615         row=12, column=4, sticky="W")
616     #=====
617     # create separator
618     sep11 = ttk.Separator(deliverables_frame, orient="horizontal")
619     sep11.grid(row=13, column=3, columnspan=2, sticky="ew")
620     #=====
621     Custom_1 = tkinter.Label(deliverables_frame, text="Ctmz_1", fg="grey").grid(row=14,
column=3)
622     Custom_1 = ttk.Combobox(deliverables_frame).grid(row=14, column=4)
623
624     Custom_2 = tkinter.Label(deliverables_frame, text="Ctmz_2", fg="grey").grid(row=15,
column=3)
625     Custom_2 = ttk.Combobox(deliverables_frame).grid(row=15, column=4)
626
627     Custom_3 = tkinter.Label(deliverables_frame, text="Ctmz_3", fg="grey").grid(row=16,
column=3)
628     Custom_3 = ttk.Combobox(deliverables_frame).grid(row=16, column=4)
629
630     Custom_4 = tkinter.Label(deliverables_frame, text="Ctmz_4", fg="grey").grid(row=17,
column=3)
631     Custom_4 = ttk.Combobox(deliverables_frame).grid(row=17, column=4)
632
633     Custom_5 = tkinter.Label(deliverables_frame, text="Ctmz_5", fg="grey").grid(row=18,
column=3)
634     Custom_5 = ttk.Combobox(deliverables_frame).grid(row=18, column=4)
635
636     #=====
637     # description box
638     #txt_box = tkinter.Text(deliverables_frame, width=65, height= 3)
639     #txt_box.grid(row=9, rowspan=10, columnspan=7, padx=1)
640
641     #txt_box_btn = tkinter.Frame(deliverables_frame)
642     #txt_box_btn = tkinter.Button(deliverables_frame, text="clear screen",
command=clear)
643     #txt_box_btn.grid(row=13, column=0)
644
645     #get_text_btn = tkinter.Frame(deliverables_frame)
646     #get_text_btn = tkinter.Button(deliverables_frame, text="Get Text",

```

```

647         command="Get_Text")
648         #get_text_btn.grid(row=13, column=1)
649     #=====
650     #=====
651     for widget in deliverables_frame.winfo_children():
652         widget.grid_configure(padx=3, pady=2)
653
654     #=====disciplines=====
655     #=====
656     global discipline_0, discipline_1, discipline_2, discipline_3, discipline_4,
657     discipline_5, discipline_6, discipline_7, discipline_8, discipline_9, discipline_10,
658     discipline_11, discipline_12, discipline_13, discipline_14, discipline_15
659     discipline_0, discipline_1, discipline_2, discipline_3, discipline_4, discipline_5,
660     discipline_6, discipline_7, discipline_8, discipline_9, discipline_10, discipline_11,
661     discipline_12, discipline_13, discipline_14, discipline_15 = tk.StringVar(
662     ), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk
663     .StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.
664     StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar(), tk.StringVar()
665
666     # folder disciplines
667     #=====
668     disciplines_frame = tkinter.LabelFrame(
669     main_frame, text="Disciplines Associated to Deliverables", relief=RIDGE)
670     disciplines_frame.grid(row=0, column=4, sticky="news", padx=3, pady=3)
671     #=====
672     #create Disciplines
673     phases_Diliverables = tkinter.Label(
674     disciplines_frame, text="D-A-to-D", fg="grey").grid(row=0, column=0, sticky="w")
675
676     #=====
677     outroselementos = tkinter.Label(
678     disciplines_frame, text="00_Ou", fg="grey").grid(row=1, column=0, sticky="w")
679     outroselementos = ttk.Combobox(disciplines_frame, values=disciplines[0], textvariable
680     =discipline_0).grid(
681     row=1, column=1, padx=3, pady=3, sticky="w")
682
683     ap_arqui = tkinter.Label(disciplines_frame, text="AP_Ar", fg="grey").grid(
684     row=2, column=0, sticky="w")
685     ap_arqui = ttk.Combobox(disciplines_frame, values=disciplines[1], textvariable=
686     discipline_1).grid(
687     row=2, column=1, padx=3, pady=3, sticky="w")
688
689     aq = tkinter.Label(disciplines_frame, text="AQ_Ar",
690     fg="grey").grid(row=3, column=0, sticky="w")
691     aq = ttk.Combobox(disciplines_frame, values=disciplines[2], textvariable=discipline_2
692     ).grid(
693     row=3, column=1, padx=3, pady=3, sticky="w")
694
695     CTtt = tkinter.Label(disciplines_frame, text="CT-Co",
696     fg="grey").grid(row=4, column=0, sticky="w")
697     CTtt = ttk.Combobox(disciplines_frame, values=disciplines[3], textvariable=
698     discipline_3).grid(
699     row=4, column=1, padx=3, pady=3, sticky="w")
700
701     EAa = tkinter.Label(disciplines_frame, text="EA-Ac",
702     fg="grey").grid(row=5, column=0, sticky="w")
703     EAa = ttk.Combobox(disciplines_frame, values=disciplines[4], textvariable=
704     discipline_4).grid(row=5,
705
706     column=1, padx=3, pady=3, sticky="w")
707
708     EEe = tkinter.Label(disciplines_frame, text="EE-Es",

```

```

696         fg="grey").grid(row=6, column=0, sticky="w")
697 EEe = ttk.Combobox(disciplines_frame, values=disciplines[5], textvariable=
discipline_5).grid(row=6,
698         column=1, padx=3, pady=3, sticky="w")
699
700 Eletr = tkinter.Label(disciplines_frame, text="EL-El",
701         fg="grey").grid(row=7, column=0, sticky="w")
702 Eletr = ttk.Combobox(disciplines_frame, values=disciplines[6], textvariable=
discipline_6).grid(
703         row=7, column=1, padx=3, pady=3, sticky="w")
704
705 GAGas = tkinter.Label(disciplines_frame, text="GA-Ga",
706         fg="grey").grid(row=8, column=0, sticky="w")
707 GAGas = ttk.Combobox(disciplines_frame, values=disciplines[7], textvariable=
discipline_7).grid(row=8,
708         column=1, padx=3, pady=3, sticky="w")
709
710 hahidr = tkinter.Label(disciplines_frame, text="HA-Hi",
711         fg="grey").grid(row=9, column=0, sticky="w")
712 hahidr = ttk.Combobox(
disciplines_frame, values=disciplines[8], textvariable=discipline_8).grid(row=9,
713         column=1, padx=3, pady=3)
714
715 mc_Mec = tkinter.Label(disciplines_frame, text="HP-Hid",
716         fg="grey").grid(row=10, column=0, sticky="w")
717 mc_Mec = ttk.Combobox(
disciplines_frame, values=disciplines[9], textvariable=discipline_9).grid(row=10,
718         column=1, padx=3, pady=3)
719
720 mv_MeV = tkinter.Label(disciplines_frame, text="MC-Mec",
721         fg="grey").grid(row=11, column=0, sticky="w")
722 mv_MeV = ttk.Combobox(
disciplines_frame, values=disciplines[10], textvariable=discipline_10).grid(row=
723         11, column=1, padx=3, pady=3)
724
725 si_SgCtl = tkinter.Label(disciplines_frame, text="MV-Mec",
726         fg="grey").grid(row=12, column=0, sticky="w")
727 si_SgCtl = ttk.Combobox(
disciplines_frame, values=disciplines[11], textvariable=discipline_11).grid(row=
728         12, column=1, padx=3, pady=3)
729
730 cc_CiCl = tkinter.Label(disciplines_frame, text="SI-Seg",
731         fg="grey").grid(row=13, column=0, sticky="w")
732 cc_CiCl = ttk.Combobox(
disciplines_frame, values=disciplines[12], textvariable=discipline_12).grid(row=
733         13, column=1, padx=3, pady=3)
734
735 CityCouncil = tkinter.Label(
disciplines_frame, text="CC-Cit", fg="grey").grid(row=14, column=0, sticky="w")
736 CityCouncil = ttk.Combobox(
disciplines_frame, values=disciplines[13], textvariable=discipline_13).grid(row=
737         14, column=1, padx=3, pady=3)
738
739
740 EId = tkinter.Label(disciplines_frame, text="EI-Ited",
741         fg="grey").grid(row=15, column=0, sticky="w")
742 EId = ttk.Combobox(disciplines_frame, values=disciplines[14], textvariable=
discipline_14).grid(
743         row=15, column=1, padx=3, pady=3)
744
745 blnk = tkinter.Label(disciplines_frame, text="Ctmz_1",
746         fg="grey").grid(row=16, column=0, sticky="w")
747 blnk = ttk.Combobox(disciplines_frame, values=disciplines[15], textvariable=
discipline_15).grid(
748         row=16, column=1, padx=3, pady=3)
749

```

```

750     blnk1 = tkinter.Label(disciplines_frame, text="Ctmz_2",
751                           fg="grey").grid(row=17, column=0, sticky="w")
752     blnk1 = ttk.Combobox(disciplines_frame, values=disciplines[15], textvariable=
discipline_15).grid(
753         row=17, column=1, padx=3, pady=3)
754
755
756     #=====disciplines option
2=====#
757
758     # disciplines_frame2 = tkinter.LabelFrame(main_frame, text="Disciplines Associated
to Deliverables Alternative", relief=RIDGE)
759     # disciplines_frame2.grid(row=0, column=5, sticky="news", padx=3, pady=3)
760
761
762     # ee1 = tkinter.Label(disciplines_frame2, text="0_1", fg="grey").grid(row=0,
column=0, sticky="w")
763     # d1 = ttk.Combobox(disciplines_frame2, values=disciplines[1],
textvariable=discipline_0, width=50).grid(row=0, column=1, padx=1, pady=3,
sticky="w")
764
765     # d2 = tkinter.Label(disciplines_frame2, text="0_2", fg="grey").grid(row=2,
column=0, sticky="w")
766     # d2 = ttk.Combobox(disciplines_frame2, width=50).grid(row=2, column=1, padx=1,
pady=3, sticky="w")
767
768
769
770     # d3 = tkinter.Label(disciplines_frame2, text="1_1", fg="grey").grid(row=3,
column=0, sticky="w")
771     # d3 = ttk.Combobox(disciplines_frame2, width=50).grid(row=3, column=1, padx=1,
pady=3, sticky="w")
772
773     # d4 = tkinter.Label(disciplines_frame2, text="1_2", fg="grey").grid(row=4,
column=0, sticky="w")
774     # d4 = ttk.Combobox(disciplines_frame2, width=50).grid(row=4, column=1, padx=1,
pady=3, sticky="w")
775
776     # d5 = tkinter.Label(disciplines_frame2, text="1_3", fg="grey").grid(row=5,
column=0, sticky="w")
777     # d5 = ttk.Combobox(disciplines_frame2, width=50).grid(row=5, column=1, padx=1,
pady=3, sticky="w")
778
779     # d6 = tkinter.Label(disciplines_frame2, text="1_4", fg="grey").grid(row=6,
column=0, sticky="w")
780     # d6 = ttk.Combobox(disciplines_frame2, width=50).grid(row=6, column=1, padx=1,
pady=3, sticky="w")
781
782     # d7 = tkinter.Label(disciplines_frame2, text="1_5", fg="grey").grid(row=7,
column=0, sticky="w")
783     # d7 = ttk.Combobox(disciplines_frame2, width=50).grid(row=7, column=1, padx=1,
pady=3, sticky="w")
784
785
786
787     # d8 = tkinter.Label(disciplines_frame2, text="2_1", fg="grey").grid(row=8,
column=0, sticky="w")
788     # d8 = ttk.Combobox(disciplines_frame2, width=50).grid(row=8, column=1, padx=1,
pady=3, sticky="w")
789
790     # d9 = tkinter.Label(disciplines_frame2, text="2_1", fg="grey").grid(row=9,
column=0, sticky="w")
791     # d9 = ttk.Combobox(disciplines_frame2, width=50).grid(row=9, column=1, padx=1,
pady=3, sticky="w")
792
793     # d10 = tkinter.Label(disciplines_frame2, text="3_1", fg="grey").grid(row=10,
column=0, sticky="w")

```

```

794     # d10 = ttk.Combobox(disциплины_frame2, width=50).grid(row=10, column=1, padx=1,
795     pady=3, sticky="w")
796     # d11 = tkinter.Label(disциплины_frame2, text="3_2", fg="grey").grid(row=11,
797     column=0, sticky="w")
798     # d11 = ttk.Combobox(disциплины_frame2, width=50).grid(row=11, column=1, padx=1,
799     pady=3, sticky="w")
800     # d12 = tkinter.Label(disциплины_frame2, text="4_1", fg="grey").grid(row=12,
801     column=0, sticky="w")
802     # d12 = ttk.Combobox(disциплины_frame2, width=50).grid(row=12, column=1, padx=1,
803     pady=3, sticky="w")
804     # d13 = tkinter.Label(disциплины_frame2, text="5_1", fg="grey").grid(row=13,
805     column=0, sticky="w")
806     # d13 = ttk.Combobox(disциплины_frame2, width=50).grid(row=13, column=1, padx=1,
807     pady=3, sticky="w")
808     # d14 = tkinter.Label(disциплины_frame2, text="5_2", fg="grey").grid(row=14,
809     column=0, sticky="w")
810     # d14 = ttk.Combobox(disциплины_frame2, width=50).grid(row=14, column=1, padx=1,
811     pady=3, sticky="w")
812     # d15 = tkinter.Label(disциплины_frame2, text="5_3", fg="grey").grid(row=15,
813     column=0, sticky="w")
814     # d15 = ttk.Combobox(disциплины_frame2, width=50).grid(row=15, column=1, padx=1,
815     pady=3, sticky="w")
816     # d16 = tkinter.Label(disциплины_frame2, text="6_1", fg="grey").grid(row=16,
817     column=0, sticky="w")
818     # d16 = ttk.Combobox(disциплины_frame2, width=50).grid(row=16, column=1, padx=1,
819     pady=3, sticky="w")
820     # d17 = tkinter.Label(disциплины_frame2, text="6_2", fg="grey").grid(row=17,
821     column=0, sticky="w")
822     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=17, column=1, padx=1,
823     pady=3, sticky="w")
824     # d17 = tkinter.Label(disциплины_frame2, text="7_1", fg="grey").grid(row=18,
825     column=0, sticky="w")
826     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=18, column=1, padx=1,
827     pady=3, sticky="w")
828     # d17 = tkinter.Label(disциплины_frame2, text="8_1", fg="grey").grid(row=19,
829     column=0, sticky="w")
830     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=19, column=1, padx=1,
831     pady=3, sticky="w")
832     # d17 = tkinter.Label(disциплины_frame2, text="8_2", fg="grey").grid(row=20,
833     column=0, sticky="w")
834     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=20, column=1, padx=1,
835     pady=3, sticky="w")
836     # d17 = tkinter.Label(disциплины_frame2, text="9_1", fg="grey").grid(row=21,
837     column=0, sticky="w")
838     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=21, column=1, padx=1,
839     pady=3, sticky="w")
840     # d17 = tkinter.Label(disциплины_frame2, text="9_2", fg="grey").grid(row=21,
841     column=0, sticky="w")
842     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=21, column=1, padx=1,
843     pady=3, sticky="w")
844     # d17 = tkinter.Label(disциплины_frame2, text="9_3", fg="grey").grid(row=21,
845     column=0, sticky="w")
846     # d17 = ttk.Combobox(disциплины_frame2, width=50).grid(row=21, column=1, padx=1,
847     pady=3, sticky="w")

```

```

834
835     # d17 = tkinter.Label(disciplines_frame2, text="9_4", fg="grey").grid(row=21,
      column=0, sticky="w")
836     # d17 = ttk.Combobox(disciplines_frame2, width=50).grid(row=21, column=1, padx=1,
      pady=3, sticky="w")
837
838     # d17 = tkinter.Label(disciplines_frame2, text="9_5", fg="grey").grid(row=21,
      column=0, sticky="w")
839     # d17 = ttk.Combobox(disciplines_frame2, width=50).grid(row=21, column=1, padx=1,
      pady=3, sticky="w")
840
841     # d17 = tkinter.Label(disciplines_frame2, text="9_6", fg="grey").grid(row=21,
      column=0, sticky="w")
842     # d17 = ttk.Combobox(disciplines_frame2, width=50).grid(row=21, column=1, padx=1,
      pady=3, sticky="w")
843
844
845     #=====button=====
      =====#
846
847     button_frame = tkinter.LabelFrame(main_frame, text="_Button_")
848     button_frame.grid(row=3, columnspan=6, sticky="news", padx=3, pady=3)
849
850     text_btn = tkinter.Frame(button_frame)
851     text_btn = tkinter.Button(
852         button_frame, text="Create folders", command=create_folders)
853     text_btn.pack(pady=5)
854
855
856     #=====button=====
      =====#
857
858
859
860     for widget in main_frame.winfo_children():
861         widget.grid_configure(padx=7, pady=2)
862
863     root.mainloop()
864

```

Appendix 2: Code used for solution 2

```

1  from array import array
2  from itertools import combinations
3  import tkinter
4  from optparse import Values
5  import select
6  from tkinter.font import BOLD
7  import tkinter.messagebox
8  import tkinter as tk
9  from tkinter import BOTTOM, E, GROOVE, LEFT, RIDGE, RIGHT, TOP, W, Button, Label,
   StringVar, Tk, ttk
10 from turtle import st, title
11 from unittest import result
12 from urllib.robotparser import RobotFileParser
13 import tkinter as tk
14 import pyodbc
15 import os
16 from tkinter import *
17 from datetime import date
18 #=====
19 ==#
20 #=====
21 ==#
22
23 class library:
24     def __init__(self, root,):
25         self.root = root
26         self.root.title("container naming System")
27         # self.root.geometry("1350x759+0+0")
28         # root.geometry("1600x1400")
29         # root.iconbitmap(
30             # "C:/Users/KNIGHT_PC/Downloads/Thesis-dev/logo.ico")
31         root.iconbitmap(
32             "C:/Users/KNIGHT_PC/Downloads/Thesis-dev/logo.ico")
33         self.project = tkinter.StringVar()
34         self.originator = tkinter.StringVar()
35         self.volume_System = tkinter.StringVar()
36         self.level_Location = tkinter.StringVar()
37         self.level_Location = tkinter.StringVar()
38         self.Type = tkinter.StringVar()
39         self.role = tkinter.StringVar()
40         self.status = tkinter.StringVar()
41         self.revision = tkinter.StringVar()
42         self.classification = tkinter.StringVar()
43         self.dates = tkinter.StringVar()
44         self.author = tkinter.StringVar()
45         self.number = tkinter.StringVar()
46
47         #=====
48         =====#
49
50         conn = pyodbc.connect(
51             r'Driver={Microsoft Access Driver (*.mdb,
52             *.accdb)};DBQ=C:/Users/KNIGHT_PC/Downloads/Thesis-dev/DB.accdb;')
53         # conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb,
54             # *.accdb)};DBQ=C:/Users/KNIGHT_PC/Downloads/Thesis-dev/DB.accdb;')
55         cursor = conn.cursor()
56         projecto = "SELECT * FROM PROJECTO"
57         cursor.execute(projecto)
58         projecto_desc = []
59         projecto_code = []
60         for row in cursor.fetchall():
61             projecto_desc.append(row.Description)
62             projecto_code.append(row.CÓDIGO)

```

```

61         origem = "SELECT * FROM ORIGEM"
62         cursor.execute(origem)
63         origem_origin = []
64         origem_code = []
65         for row in cursor.fetchall():
66             origem_origin.append(row.ORIGINATOR)
67             origem_code.append(row.CÓDIGO)
68
69         volume_sistema = "SELECT * FROM VOLUME_SISTEMA"
70         cursor.execute(volume_sistema)
71         vsistema = []
72         vsistema_code = []
73         for row in cursor.fetchall():
74             vsistema.append(row.VOLUME_SISTEMA)
75             vsistema_code.append(row.CÓDIGO)
76
77         nivel = "SELECT * FROM NÍVEL_PISO"
78         cursor.execute(nivel)
79         nivel_piso = []
80         nivel_piso_code = []
81         for row in cursor.fetchall():
82             nivel_piso.append(row.NÍVEL_PISO)
83             nivel_piso_code.append(row.CÓDIGO)
84
85         tipogen = "SELECT * FROM TIPO"
86         cursor.execute(tipogen)
87         tipo = []
88         tipo_code = []
89         for row in cursor.fetchall():
90             tipo.append(row.TIPO)
91             tipo_code.append(row.CÓDIGO)
92
93         disciplina = "SELECT * FROM DISCIPLINA_FUNÇÃO"
94         cursor.execute(disciplina)
95         disciplina_func = []
96         disciplina_code = []
97         for row in cursor.fetchall():
98             disciplina_func.append(row.DISCIPLINA_FUNÇÃO)
99             disciplina_code.append(row.CÓDIGO)
100
101         stat = "SELECT * FROM STATUS"
102         cursor.execute(stat)
103         status = []
104         status_code = []
105         # estado = []
106         for row in cursor.fetchall():
107             status.append(row.STATUS)
108             status_code.append(row.CÓDIGO)
109             # estado.append(row.ESTADO)
110
111         rev = "SELECT * FROM REVISÃO"
112         cursor.execute(rev)
113         revisao = []
114         revisao_code = []
115         for row in cursor.fetchall():
116             revisao.append(row.REVISÃO)
117             revisao_code.append(row.CÓDIGO)
118
119         uni = "SELECT * FROM UniClass2015"
120         cursor.execute(uni)
121         classification = []
122         classification_code = []
123         for row in cursor.fetchall():
124             classification.append(row.Title)
125             classification_code.append(row.Code)
126
127         #
128         est = "SELECT * FROM ESTADO"

```

```

128         cursor.execute(est)
129         estado = []
130         estado_code = []
131         for row in cursor.fetchall():
132             estado.append(row.ESTADO)
133             estado_code.append(row.CÓDIGO)
134
135         est = "SELECT * FROM AUTHOR"
136         cursor.execute(est)
137         autor = []
138         autor_code = []
139         for row in cursor.fetchall():
140             autor.append(row.AUTOR)
141             autor_code.append(row.CÓDIGO)
142
143         ft = "SELECT * FROM FILE_TYPE"
144         cursor.execute(ft)
145         file_type = []
146         for row in cursor.fetchall():
147             file_type.append(row.Formats)
148     # Functions
149
150     def add_to_db():
151         try:
152             for i in enumerate(idlist.get(0, END)):
153                 x = i[1]
154                 values = x.split("|")
155                 P1 = values[0]
156                 P2 = values[1]
157                 P3 = values[2]
158                 P4 = values[3]
159                 P5 = values[4]
160                 P6 = values[5]
161                 P7 = values[6]
162                 P8 = values[7]
163                 P9 = values[8]
164                 P10 = values[9]
165                 P11 = values[10]
166                 P12 = values[11]
167                 P13 = values[12]
168                 P14 = values[13]
169                 P15 = values[14]
170                 db = ((P1, P2, P3, P4, P5, P6, P7, P8,
171                     P9, P10, P11, P12, P13, P14, P15))
172                 cursor = conn.cursor()
173                 cursor.execute(
174                     '''INSERT INTO ContainerName (Project, Originator,
175                     Volume_System, Level_Location, Types, Roles, Numbers, Status,
176                     Revision, Classification, States, Author, Dates, File_Type,
177                     Descriptions) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
178                     ?)''' , db)
179                 conn.commit()
180                 tkinter.messagebox.showinfo(
181                     'Tips:', 'Save Successful!')
182                 idlist.delete(0, END)
183                 self.Label2.delete(0, END)
184             except:
185                 tkinter.messagebox.showerror(
186                     'Tips', 'Something Went Wrong')
187
188     def create_id_combination(self, array_value):
189         self.Label2.delete(0, END)
190         result1 = find_code(projecto_desc, projecto_code, state1)
191         result2 = find_code(origem_origin, origem_code, state2)
192         result3 = find_code(vsistema, vsistema_code, state3)
193         result4 = find_code(nivel_piso, nivel_piso_code, state4)
194         result5 = find_code(tipo, tipo_code, state5)

```

```

191         result6 = find_code(disciplina_func, disciplina_code, state6)
192         result7 = state7.get()
193         result8 = find_code(status, status_code, state8)
194         result9 = find_code(revisao, revisao_code, state9)
195         result10 = find_code(classification, classification_code, state10)
196         result11 = find_code(estado, estado_code, state11)
197         result12 = find_code(autor, autor_code, state12)
198         result13 = state13.get()
199         result14 = state14.get()
200         result15 = state15.get()
201         global db_values
202         array_value = [result1, result2, result3, result4, result5, result6,
203                        result7, result8, result9, result13, result15]
204         db_values = [result1, result2, result3, result4, result5, result6,
205                     result7, result8, result9, result10, result11, result12,
206                     result13, result14, result15]
207
208         if len(result15) > 20:
209             tkinter.messagebox.showerror(
210                 'Tips', 'Description is more than the limit')
211         else:
212             try:
213                 while True:
214                     array_value.remove(None)
215             except ValueError:
216                 pass
217             try:
218                 i = 0
219                 db_values = ["None" if v is None else v for v in db_values]
220             except ValueError:
221                 pass
222             try:
223                 while True:
224                     array_value.remove("")
225             except ValueError:
226                 pass
227             global combination, details
228             combination = "-"
229             details = "|"
230             combination = combination.join(array_value)
231             details = details.join(db_values)
232             combination = combination + result14
233             if state14.get() == "":
234                 self.Label2.delete(0, END)
235                 tkinter.messagebox.showerror(
236                     'Tips', 'File Type is not specified')
237             else:
238                 self.Label2.insert(0, combination)
239                 tkinter.messagebox.showinfo(
240                     'Tips:', 'Done!')
241
242         def clear_id():
243             self.Label.delete(0, END)
244
245         def clear():
246             idlist.delete(0, END)
247
248         def add_id(value):
249             idlist.insert(0, value)
250
251         def copy_id():
252             self.root.clipboard_clear()
253             self.root.clipboard_append(combination)
254             tkinter.messagebox.showinfo('Tips:', 'Copied!')
255
256         def display_id(self, value):
257             self.Label.insert(0, value)

```

```

257     def find_code(main, code, state):
258         if state.get() != "":
259             i = main.index(state.get())
260             value = code[i]
261             return value
262
263         #=====frame=====
264         #
265         # def select():
266         #prj = tkinter.Entry(rcrds_Label, text=prjcombobox.get())
267         global statel, state2, state3, state4, state5, state6, state7, state8, state9,
268         statel0, statel1, statel2, statel3,
269         statel, state2, state3, state4, state5, state6, state7, state8, state9, statel0,
270         statel1, statel2, statel3, statel4, statel5 = tkinter.StringVar(), tkinter.
271         StringVar(), tkinter.StringVar(), tkinter.StringVar(),
272         ), tkinter.StringVar(), tkinter.StringVar(), tkinter.StringVar(), tkinter.
273         StringVar(), tkinter.StringVar(), tkinter.StringVar(), tkinter.StringVar(),
274         tkinter.StringVar(), tkinter.StringVar(), tkinter.StringVar(), tkinter.StringVar
275         ()
276         today = [date.today()]
277         global array
278         array = [statel.get(), state2.get(), state3.get(), state4.get(), state5.get(),
279         state6.get(
280         ), state7.get(), state8.get(), state9.get(), statel0.get(), statel1.get(),
281         statel2.get(), statel3.get(), statel4.get()]
282         mainframe = tkinter.Frame(self.root)
283         mainframe.pack(padx=5, pady=5)
284
285         rcrds_Label = tkinter.LabelFrame(
286             mainframe, text="ID Attributes", padx=5, pady=5)
287         rcrds_Label.grid(row=0, column=0, sticky="news")
288
289         self.prj = tkinter.Label(rcrds_Label, text="Project",).grid(
290             row=0, column=0, padx=5, pady=5)
291         self.prj = ttk.Combobox(rcrds_Label, values=projecto_desc, textvariable=statel).
292         grid(
293             row=0, column=1, padx=5, pady=5)
294
295         self.org = tkinter.Label(rcrds_Label, text="Originator",).grid(
296             row=0, column=2, padx=5, pady=5)
297         self.org = ttk.Combobox(rcrds_Label, values=origem_origin, textvariable=state2).
298         grid(
299             row=0, column=3, padx=5, pady=5)
300
301         self.vol = tkinter.Label(rcrds_Label, text="Volume_System",).grid(
302             row=0, column=4, padx=5, pady=5)
303         self.vol = ttk.Combobox(rcrds_Label, values=vsistema, textvariable=state3).grid(
304             row=0, column=5, padx=5, pady=5)
305
306         self.ll = tkinter.Label(rcrds_Label, text="Level_Location",).grid(
307             row=0, column=6, padx=5, pady=5)
308         self.ll = ttk.Combobox(rcrds_Label, values=nivel_piso, textvariable=state4).grid(
309             row=0, column=7, padx=5, pady=5)
310
311         self.ty = tkinter.Label(rcrds_Label, text="Type",).grid(
312             row=0, column=8, padx=5, pady=5)
313         self.ty = ttk.Combobox(rcrds_Label, values=tipo, textvariable=state5).grid(
314             row=0, column=9, padx=5, pady=5)
315
316         self.rl = tkinter.Label(rcrds_Label, text="Role",).grid(
317             row=0, column=10, padx=5, pady=5)
318         self.rl = ttk.Combobox(rcrds_Label, values=disciplina_func, textvariable=state6).
319         grid(
320             row=0, column=11, padx=5, pady=5)

```

```

311     self.nbr = tkinter.Label(rcrds_Label, text="Number",).grid(
312         row=0, column=12, padx=5, pady=5)
313     self.nbr = ttk.Spinbox(rcrds_Label, from_="000001", to="infinity", textvariable=
state7).grid(
314         row=0, column=13, padx=5, pady=5)
315
316     #=====
317
318     #=====
319
320     metadata = tkinter.LabelFrame(
321         mainframe, text="MetaData Attributues", padx=5, pady=5)
322     metadata.grid(row=1, column=0, sticky="news")
323
324     self.sts = tkinter.Label(metadata, text="Status",).grid(
325         row=0, column=0, padx=5, pady=5)
326     self.sts = ttk.Combobox(metadata, values=status, textvariable=state8).grid(
327         row=0, column=1, padx=5, pady=5)
328
329     self.rev = tkinter.Label(metadata, text="Revision",).grid(
330         row=0, column=2, padx=5, pady=5)
331     self.rev = ttk.Combobox(metadata, values=revisao, textvariable=state9).grid(
332         row=0, column=3, padx=5, pady=5)
333
334     self.cl = tkinter.Label(metadata, text="Classification",).grid(
335         row=0, column=4, padx=5, pady=5)
336     self.cl = ttk.Combobox(metadata, values=classification, textvariable=state10).
grid(
337         row=0, column=5, padx=5, pady=5)
338
339     self.sta = tkinter.Label(metadata, text="States",).grid(
340         row=0, column=6, padx=5, pady=5)
341     self.sta = ttk.Combobox(metadata, values=estado, textvariable=state11).grid(
342         row=0, column=7, padx=5, pady=5)
343
344     self.aut = tkinter.Label(metadata, text="Author",).grid(
345         row=0, column=8, padx=5, pady=5)
346     self.aut = ttk.Combobox(metadata, values=autor, textvariable=state12).grid(
347         row=0, column=9, padx=5, pady=5)
348
349     self.dt = tkinter.Label(metadata, text="Date",).grid(
350         row=0, column=10, padx=5, pady=5)
351     self.dt = ttk.Combobox(metadata, values=today, textvariable=state13).grid(
352         row=0, column=11, padx=5, pady=5)
353
354     self.ft = tkinter.Label(metadata, text="File Type",).grid(
355         row=0, column=13, padx=5, pady=5)
356     self.ft = ttk.Combobox(metadata, values=file_type, textvariable=state14).grid(
357         row=0, column=14, padx=5, pady=5)
358
359     #=====
360     self.icic = tkinter.LabelFrame(
361         mainframe, text="Description/Information Container Identification Code")
362     self.icic.grid(row=2, column=0, sticky="news")
363
364     self.Label1 = tkinter.Label(
365         self.icic, text="Description").grid(row=0, column=0)
366     self.Label1 = tk.Entry(self.icic, width=100, textvariable=state15)
367     self.Label1.grid(row=0, column=1, columnspan=3, pady=5, padx=10)
368
369     self.sep001 = ttk.Separator(self.icic, orient="vertical").grid(

```

```

370         row=0, column=4, padx=3, pady=5, sticky="ns")
371
372     self.Label2 = tkinter.Label(self.icic, text="ID", font="bold").grid(
373         row=0, padx=3, column=5, sticky="e")
374     self.Label2 = tk.Entry(self.icic, width=130)
375     self.Label2.grid(row=0, column=6)
376
377
378     #=====
379     buttonframe = tkinter.LabelFrame(
380         mainframe, text="Button Frame", width=300, height=50, padx=1, pady=2)
381     buttonframe.grid(row=4, column=0, sticky="news")
382
383     addbbtn1 = tkinter.Button(
384         buttonframe, text="ADD ID", width=10, height=1, command=lambda: add_id(
            details))
385     addbbtn1.grid(row=0, column=0)
386
387     addbbtn3 = tkinter.Button(
388         buttonframe, text="CLEAR LIST", width=10, height=1, command=lambda: clear())
389     addbbtn3.grid(row=0, column=3)
390
391     addbbtn4 = tkinter.Button(
392         buttonframe, text="COPY ID", width=10, height=1, command=lambda: copy_id())
393     addbbtn4.grid(row=0, column=4)
394
395     addbbtn5 = tkinter.Button(
396         buttonframe, text="CREATE ID COMBINATION", width=20, height=1, command=lambda
            : create_id_combination(self, array))
397     addbbtn5.grid(row=0, column=5)
398
399     addbbtn6 = tkinter.Button(
400         buttonframe, text="ADD TO DATABASE", width=20, height=1, command=lambda:
            add_to_db())
401     addbbtn6.grid(row=0, column=6)
402
403     #=====list box and
404     scrollbar=====#
405
406     self.prt_ID_list = tkinter.LabelFrame(
407         mainframe, text="Project ID Data List", padx=5, pady=5)
408     self.prt_ID_list.grid(row=5, column=0, sticky="news")
409
410     scrollbar = tkinter.Scrollbar(self.prt_ID_list)
411     scrollbar.grid(row=0, column=1, sticky="ns")
412
413     idlist = tkinter.Listbox(
414         self.prt_ID_list, width=250, height=10, yscrollcommand=scrollbar.set)
415     idlist.grid(row=0, column=0, sticky="new")
416     scrollbar.config(command=idlist.yview)
417
418     #=====
419
420     for widget in mainframe.winfo_children():
421         widget.grid_configure(padx=10, pady=15, sticky="news")
422
423 if __name__ == '__main__':
424     root = Tk()
425     application = library(root)
426     root.mainloop()

```

Appendix 3: Code used for solution 3

```

1  from array import array
2  from itertools import combinations
3  import tkinter
4  from optparse import Values
5  import select
6  from tkinter.font import BOLD
7  from tkinter.filedialog import askdirectory
8  import tkinter.messagebox
9  import tkinter as tk
10 from tkinter import BOTTOM, E, GROOVE, LEFT, RIDGE, RIGHT, TOP, W, Button, Label,
    StringVar, Tk, ttk
11 from turtle import st, title
12 from unittest import result
13 from urllib.robotparser import RobotFileParser
14 import tkinter as tk
15 import pyodbc
16 import os
17 from tkinter import *
18 from datetime import datetime
19
20
21 #=====#
22
23
24 root = tkinter.Tk()
25 root.title("Information Verification")
26 root.iconbitmap(
27     "C:/Users/KNIGHT_PC/Downloads/Thesis-dev/logo.ico")
28
29 conn = pyodbc.connect(
30     r'Driver={Microsoft Access Driver (*.mdb,
31     *.accdb)};DBQ=C:/Users/KNIGHT_PC/Downloads/Thesis-dev/DB.accdb;')
32 cursor = conn.cursor()
33 projecto = "SELECT * FROM PROJECTO"
34 cursor.execute(projecto)
35 projecto_code = []
36 for row in cursor.fetchall():
37     projecto_code.append(row.CÓDIGO)
38
39 origem = "SELECT * FROM ORIGEM"
40 cursor.execute(origem)
41 origem_code = []
42 for row in cursor.fetchall():
43     origem_code.append(row.CÓDIGO)
44
45 volume_sistema = "SELECT * FROM VOLUME_SISTEMA"
46 cursor.execute(volume_sistema)
47 vsistema_code = []
48 for row in cursor.fetchall():
49     vsistema_code.append(row.CÓDIGO)
50
51 nivel = "SELECT * FROM NÍVEL_PISO"
52 cursor.execute(nivel)
53 nivel_piso_code = []
54 for row in cursor.fetchall():
55     nivel_piso_code.append(row.CÓDIGO)
56
57 tipogen = "SELECT * FROM TIPO"
58 cursor.execute(tipogen)
59 tipo_code = []
60 for row in cursor.fetchall():
61     tipo_code.append(row.CÓDIGO)
62
63 disciplina = "SELECT * FROM DISCIPLINA_FUNÇÃO"
64 cursor.execute(disciplina)
65 disciplina_code = []
66 for row in cursor.fetchall():

```

```

66     disciplina_code.append(row.CÓDIGO)
67
68     stat = "SELECT * FROM STATUS"
69     cursor.execute(stat)
70     status_code = []
71     for row in cursor.fetchall():
72         status_code.append(row.CÓDIGO)
73
74     rev = "SELECT * FROM REVISÃO"
75     cursor.execute(rev)
76     revisao_code = []
77     for row in cursor.fetchall():
78         revisao_code.append(row.CÓDIGO)
79
80     ft = "SELECT * FROM FILE_TYPE"
81     cursor.execute(ft)
82     file_type = []
83     for row in cursor.fetchall():
84         file_type.append(row.Formats)
85
86     Checkbutton1 = IntVar()
87     path = tkinter.StringVar()
88
89
90     def selectPath():
91         path_ = askdirectory()
92         path.set(path_)
93
94
95     def getFiles(folder):
96         if Checkbutton1.get() == 1:
97             for path, subdirs, files in os.walk(folder):
98                 for name in files:
99                     checkConfirmation(name)
100         else:
101             tkinter.messagebox.showerror('Tips', 'Tick Checkbox To Allow verification')
102
103
104     def is_date(string):
105         format_yyyymmdd = "%Y-%m-%d"
106         try:
107             date = datetime.strptime(string, format_yyyymmdd)
108             return True
109         except ValueError:
110             return False
111
112
113     def clear():
114         my_Listbox.delete(0, END)
115         my_Listbox1.delete(0, END)
116
117
118     def checkConfirmation(file):
119         filename = os.path.basename(file)
120         splitFileName = filename.split('-')
121         if len(splitFileName) == 13:
122             n = len(splitFileName) - 1
123             date_values = [splitFileName[9], splitFileName[10], splitFileName[11]]
124             date = "-"
125             date = date.join(date_values)
126             desc_file = splitFileName[n].split('.')
127             fileType = "." + desc_file[1]
128             print(desc_file, fileType)
129             if splitFileName[0] not in projecto_code:
130                 my_Listbox1.insert(END, filename)
131             elif splitFileName[1] not in origem_code:
132                 my_Listbox1.insert(END, filename)

```

```

133         elif splitFileName[2] not in vsistema_code:
134             my_Listbox1.insert(END, filename)
135         elif splitFileName[3] not in nivel_piso_code:
136             my_Listbox1.insert(END, filename)
137         elif splitFileName[4] not in tipo_code:
138             my_Listbox1.insert(END, filename)
139         elif splitFileName[5] not in disciplina_code:
140             my_Listbox1.insert(END, filename)
141         elif isinstance(int(splitFileName[6]), int) == False:
142             my_Listbox1.insert(END, filename)
143         elif splitFileName[7] not in status_code:
144             my_Listbox1.insert(END, filename)
145         elif splitFileName[8] not in revisao_code:
146             my_Listbox1.insert(END, filename)
147         elif is_date(date) == False:
148             my_Listbox1.insert(END, filename)
149         elif fileType not in file_type:
150             my_Listbox1.insert(END, filename)
151         else:
152             my_Listbox.insert(END, filename)
153     else:
154         my_Listbox1.insert(END, filename)
155
156     #=====
157     =====#
158
159     frame1 = tkinter.LabelFrame(root)
160     folder_path = tkinter.Label(frame1, text="Folder Path:").grid(
161         row=0, column=0, sticky="w")
162     path_entry_label = tkinter.Entry(
163         frame1, width=30, textvariable=path).grid(row=0, column=1, sticky="w")
164     ver_but = tkinter.Button(frame1, text="SELECT PATH", command=selectPath).grid(
165         row=1, column=1, sticky="news", pady=2)
166
167     sep1 = ttk.Separator(frame1, orient="horizontal").grid(
168         row=2, column=0, sticky="WE", columnspan=2, pady=2)
169
170
171     check1 = tkinter.Checkbutton(frame1, text="Check Folders items for Naming Consistency",
172                                 variable=Checkbutton1,
173                                 onvalue=1,
174                                 offvalue=0).grid(row=5, column=0, sticky="w", pady=2)
175
176     ver_but = tkinter.Button(frame1, text="RUN VERIFICATION", command=lambda: getFiles(
177         path.get())).grid(row=6, columnspan=2, sticky="news", pady=5)
178
179     #=====
180     =====#
181
182     frame2 = tkinter.LabelFrame(root)
183     Conforming_Label = tkinter.Label(
184         frame2, text="Conforming Files Displayed below:").grid(row=1, column=0, sticky="w")
185
186     #=====
187     =====#
188
189     frame3 = tkinter.LabelFrame(root)
190     my_Listbox = tkinter.Listbox(frame3, width=74)
191     my_Listbox.grid(row=2, column=0, sticky="news")
192     scrollbar = Scrollbar(frame3, orient='vertical', command=my_Listbox.yview)
193     scrollbar.grid(row=2, column=1, sticky=tk.NS)
194
195     frame4 = tkinter.LabelFrame(root)
196     Conforming_Label = tkinter.Label(

```

```

197     frame4, text="Non-Conforming Files Displayed below:").grid(row=3, column=0, sticky=
198         "w")
199
200     frame5 = tkinter.LabelFrame(root)
201
202     # scrollbar1 = tkinter.Scrollbar(frame5)
203     # scrollbar1.grid(row=4, column=1, sticky="ns")
204
205     my_Listbox1 = tkinter.Listbox(frame5, width=74)
206     my_Listbox1.grid(row=4, column=0, sticky="new")
207     scrollbar1 = Scrollbar(frame5, orient='vertical', command=my_Listbox1.yview)
208     scrollbar1.grid(row=4, column=1, sticky=tk.NS)
209     #=====clear
210     button=====#
211
212     frame6 = tkinter.LabelFrame(root)
213
214     clrbutton = tkinter.Button(frame6, text="Clear ALL", command=lambda: clear())
215     clrbutton.grid(row=5, column=1, sticky="news")
216
217     gen_report = tkinter.Button(frame6, text="Generate Report")
218     gen_report.grid(row=5, column=2, sticky="news")
219     #=====spacing=====
220     ==#
221     for widget in root.winfo_children():
222         widget.grid_configure(padx=10, pady=10)
223     #
224     =====
225     =#
226     root.mainloop()

```