TÉCNICO LISBOA

DTGov: Digital Transformation of Government Business Processes

Nuno Gonçalo da Rocha Marques

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. André Ferreira Ferrão Couto e Vasconcelos

Examination Committee

Chairperson: Prof. António Manuel Ferreira Rito da Silva
Supervisor: Prof. André Ferreira Ferrão Couto e Vasconcelos
Member of the Committee: Prof. Alberto Manuel Rodrigues da Silva

November 2022

Acknowledgments

I would like to start by thanking my mother Manuela. She was the one to encourage me to pursue this career and she always helped me, cared for me, raised me, endured all the problems, and many, many more things that are too many to fit in here. She is the strongest person I know and because of all that, thank you for being the best mom possible. This thesis is for you.

I would also like to thank the remaining of my closest family, father, brother, and sister for supporting me all these years and being there when I needed the most. Also a big thank you to the rest of the family that supported me, especially the ones that saw the beginning but not the end of this journey.

I would like to acknowledge Prof. André Vasconcelos for not only accepting me and the idea of this dissertation but also because of his insight and knowledge. Also a huge thank you to VILT for proving the opportunity to realize this thesis and to help me grow.

I would also like to thank my friends that supported and helped me with personal and academic issues

Last but not least, I would like to thank my girlfriend Mariana for always staying beside me and enduring me all these years, pushing me to be a better person. Love you.

To every one that directly or indirectly helped me on this journey – Thank you.

Abstract

Today, Digital Transformation (DT) is almost mandatory for an organization to remain relevant, the government being no exception. There are several benefits such as improved services and increased satisfaction, but there are also disadvantages such as cost, time, and security. Specifically, regarding the DT of government processes, there is little information that can be used, and the problem is compounded by the fact that most of the research focuses on citizens' applications rather than the employees' ones. This dissertation aims to solve this, by providing a solution proposition divided into 3 parts: Behavioral Model, Reference Architecture (RA), and Implementation. Before that, an analysis is made of the main use cases related to the handling of governmental processes: resolve a procedure, send a notification to the citizen by hand, reclassify a procedure and create a request. Following that, it is presented the Domain Model and Lifecycles which comprises the Domain Model, regarding the entities of the main identified projects, Procedures, Common Parts, Notifications and Requests, and their life cycles. RA adds to the entities the remaining modules such as "BPMs" and web services, adding dependencies between projects and between the citizens' applications. Having mentioned that, it is presented how to implement it in the real world using "low-code" technology, especially useful to mitigate several problems that arise from this transformation. Finally, an evaluation of the proposed solution is also presented. In summary, the information presented is not new for the "experts" in the area, but regardless it is useful to mitigate errors and for the "juniors" or employees to better understand the DT made. There are several problems found, however, that should be addressed in future work such as a better evaluation of what is proposed.

Keywords

Digital Transformation; Government; Processes; Low-code; Reference Architecture; Domain Models.

Resumo

Hoje em dia, a Transformação Digital (TD) é quase obrigatória para que as organizações se mantenham relevantes, não sendo os governos exceção. Existem diversos benefícios como melhoria de serviços e aumento da satisfação mas também desvantagens como custos, tempo, e segurança. Especificamente, relativamente à TD de processos governamentais, existe pouca informação que pode ser usada, sendo o problema agravado pelo facto de que a maioria da pesquisa se concentrar na aplicação dos cidadãos e não na dos funcionários. Esta dissertação visa resolver esse problema, fornecendo uma proposta de solução divida em 3 partes: Modelos Comportamentais, Arquitetura de Referência (AR), e Implementação. Antes disso, é feita uma análise dos principais casos de uso referentes aos processos governamentais: resolver um procedimento, envio de notifição ao cidadão por mão, reclassifcar um procedimento e criar um pedido. Posteriormente, é apresentado o Modelo de Domínio e Ciclos de Vida, que estão relacionados com as entidades dos principais projectos identificados, Procedimentos, Partes Comuns, Notificações e Pedidos. A AR acrescenta às entidades, os restantes módulos como os ”BPMs” e serviços web, adicionando as dependências entre projetos e entre a aplicação dos cidadãos. Posto isto, é apresentado como proceder à sua implementação no mundo real recorrendo à tecnologia “low-code”, especialmente útil para mitigar vários problemas que surgem dessa transformação. Por último, é também apresentada uma avaliação da solução proposta. Em resumo, a informação apresentada não é nova para os “experts” na área, mas independentemente disso é útil para mitigar erros e para que os “juniors” ou funcionários entendam melhor a TD feita. Existem vários problemas encontrados, no entanto, que devem ser abordados em trabalhos futuros como a uma melhor avaliação do proposto.

Palavras Chave

Transformação Digital; Governo; Processos; Low-code; Arquitetura de Referência; Modelos de Domínio.

Contents

List of Figures

List of Tables

Acronyms

DT Digital Transformation

IT Information Technology

RAD Rapid Application Development

UML Unified Modeling Language

CWS Collaborative Workspace

G2E Government to Employee

G2C Government to Citizen

G2B Government to Business

G2G Government to Government

EIM Enterprise Information Management

SLR Systematic Literature Review

AMSTAR A MeaSurement Tool to Assess systematic Reviews

SOA Service-Oriented Architecture

BPMN Business Process Model and Notation

EIM Enterprise Information Management

RA Reference Architecture

AI Artificial Intelligence

IoT Internet of Things

RPA Robotic Process Automation

REST Representational State Transfer

SOAP Simple Object Access Protocol

BPM Business Process Model

1

Introduction

Contents

This chapter presents the context and the problem that is tackled in this document, also mentioning how it is organized.

1.1 Context

The concept of Digital Transformation (DT) is very important in today's world, referring to the "process of using digital technologies to create new - or modify existing - business processes, culture, and customer experiences to meet changing business and market requirements" [1].

Although some issues can arrive from it, such as problems regarding security [2], employee training [3] and costs [4], the consensus is that the benefits outweigh the negatives, as seen in today's biggest companies such as Amazon or Alibaba [5].

Having improved digitization of processes offers various advantages: allows faster delivery of services [6], helps with the interoperability and internet connectivity problems [7], and mitigates issues regarding corruption and improvement of "interdependencies between actors, technologies, processes" [8].

Despite the many advantages that were and will be mentioned, organizations are still reluctant to perform the DT, for example, because of the inexperience and the need for training regarding how to proceed and what to expect from that implementation, [9] and [10], or others, some already mentioned. It is especially hard in governments since they rely on having too much data and users, making security and privacy especially difficult to control in those cases [11].

One type of technology that can ease the problems mentioned is the low-code technology [12]. Low-code technologies/platforms "are a collection of tools that enable the visual development of applications through modeling and a graphical interface" enabling "developers to skip hand-coding, speeding up the process of getting an application to production" [12]. Since it focuses on the feedback aspect and leaves the complications of development, it is an exceptionally useful tool to use in this type of problem. One issue, however, is that since it is a new technology there are still few studies and other information regarding its implementation in a real-world scenario. Adding to this, since most of its use is in projects related to a specific private organization or government, privacy is mandatory, leaving that information inside the companies that perform it. This causes several issues given that each company performs the DT differently, even if the same guidelines were provided by the organization. Usage of different software, different models followed ways to operate and others. This is especially true if the organization in question is a government.

Some governments require the creation of a public contest to find the company or companies that will be responsible for its DT. Furthermore, if the development requires various steps, it can be possible that each step needs to go to a public contest again. This means that, without generalized reference

architectures or models to follow, the probability of interoperability issues increases, given that other companies need to continue the work previously done, which costs time, resources, and more (because they need to learn how and what was already made).

1.2 Problem

The term DT is very broad and it is not possible to create models or a reference architecture that comprises every single type of DT. That reason, and others, is why there are still very few models that provide information related to DT. Also, organizations and especially governments are still reluctant to perform that DT because of several issues, but mainly costs and time, which could both be mitigated if more information such as models existed or through the usage of emerging technologies.

Having that clarified, this dissertation focuses on the DT of government processes (can also be mentioned as procedures), with the objective to provide a reference architecture and models that can be used when performing it, and also an explanation on how to implement them, using low-code technologies (useful to mitigate the issues mentioned). It is important to refer that what is here presented is based on a real-world implementation of a government DT, but that information is provided as such that each government can understand and adapt what is here to their specific cases, making the necessary changes. Different government processes can have different use cases from the ones that will be presented or other entities that are not presented here. Because of that, it is important to mention that the work here presented is not final and should be improved in the future, through more studies, the improvement of the low-code technology, and more testing in other real-world scenarios.

1.3 Organization of the Document

This dissertation is organized as follows:

1. The Related Work section, which aims to provide a background and state of the art regarding the digital transformation of government processes.

2. Use Cases section that presents some of the main government use cases when dealing with their processes.

3. The Solution Proposition section that includes Domain Model and Lifecycles sub-section which contains the entity models of the various entities relevant to this type of transformation and their lifecycles, the Reference Architecture sub-section, that relates to the reference architecture created here, comprising the different parts/projects and their modules (such as web services or roles),

and the Application Implementation sub-section, that presents how to use a low-code application to implement the information previously provided.

4. Lastly, the section Evaluation contains the evaluation and discussion of the solution proposed by interviewing Juniors and Experts (in the field and that were responsible for the real-world implementation).

This dissertation finishes with the Conclusion and Future Work, containing the summary of the work presented and suggestions for future improvements.

2

Related Work

Contents

This chapter presents the related work regarding the topics of this dissertation, which relate to the digital transformation of government processes using low-code technologies. The objective is to research papers that contain those topics and try to create a state-of-the-art regarding the principal components/-modules used in them.

This chapter contains two parts. The first, the Background, explains this dissertation's main subjects and provides the motives behind its creation (through a systematic literature review). The second part, consists on the state of the art regarding the topics mentioned, finding the main modules in them, and presenting the low-code solutions to implement them (resorting to a second SLR).

2.1 Background

DT can vastly improve the organizations' processes if applied and used correctly to enhance efficiency or improve costs [3]. Nevertheless, there are a few risks associated, such as current e-government applications still not being enough [13] or limitations in the automation of highly creative processes [14].

This section presents the three main themes of this dissertation, digital transformation in government, process automation, and low-code technologies. It also presents an SLR to understand the main problems with the DT implementation, the current representation models (and if they are enough), and whether low-code approaches are being utilized or disregarded.

2.1.1 Digital Transformation

Government "X.0"

Before mentioning anything, it is relevant to start from the beginning, which means mentioning E-government generations or government "X.0"(term coined here to refer to the various e-government generations).

Very briefly, and since the focus is not on the evolution of e-government, government "X.0" comprehends from 1.0 to 5.0 [6]. Relevant here is the government 3.0 and 4.0. Government 3.0 was the automation phase, where after having a high volume and variety of services (government 1.0 and 2.0), it was necessary to focus the attention on the efficiency of the delivery [6]. This step allows the government to improve its services by reducing costs and time associated and by improving its efficiency, as already mentioned. The Government 4.0, Digitalization phase, improved the automation previously obtained by going further beyond that, by "exploring new business models and new types of customer engagement" [6].

Also important to note is the Government to Employee (G2E) which consists in "empowering their own employees to assist citizens in the fastest and most appropriate way, speed-up administrative pro-

cesses, and optimize governmental solutions. Before we can expect citizens to go online, government employees must go online" [15]. The majority of studies focus on the Government to Citizen (G2C), Government to Business (G2B) or Government to Government (G2G) and leave behind the G2E, with the aggravating that the "majority of governments have not yet begun to develop collaborative G2E services but it is certainly need to improve the bureaucracy's day-to-day functions and dealing with employees" [15].

In order to perform the DT, it is necessary to consider the relevant stakeholders, such as employees or citizens, which services will be implemented, the technology necessary to implement them, and what type of communication will be used. Emerging technologies that enhance transformation can help in that regard.

Models

There are plenty of models related to digital transformation, and each government can (and should) take advantage of them to help them plan and execute the goals proposed. One model presented, given its importance, is the value-based Digital Government Model, Figure 2.1, featuring five critical elements (data, technology, service, people, and governance).

This model was proposed by Jungwoo Lee and B. Joon Kim [16] and consists of two levels: macro and micro. "Awareness of problems at the macro-level conceptually leads to integrative solutions by converging five different but related components of digital government".

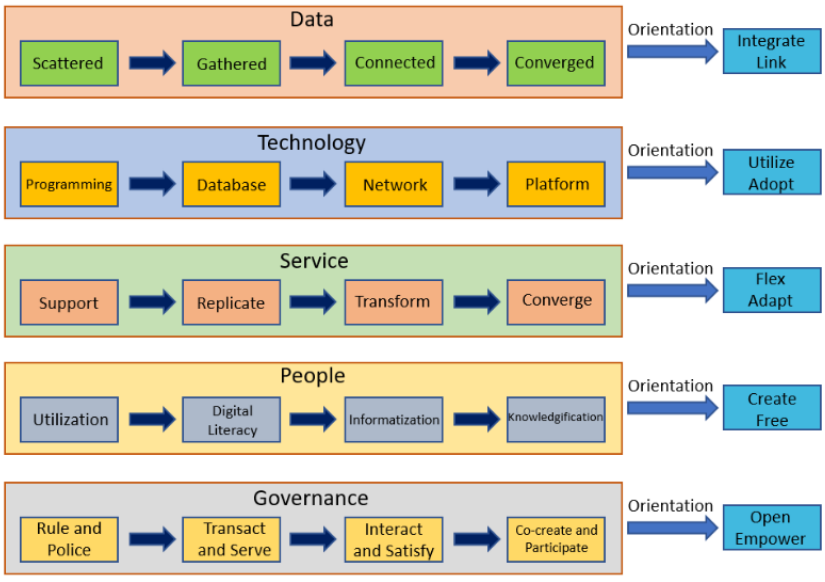

Figure 2.1: Value-based Digital Government Model(original representation)

- Data: Initial data that came dispersed from different government functions was assembled in large, locally centralized databases that afterwards were connected and converged to each other. The orientation has been to integrate and link the data from the diverse government entities.
- Technology: From simple programs to utilizing databases to store that data. After the appearance of the internet, technology as a platform became more mainstream. The orientation has been to utilize and adopt new technologies.
- Service: Systems initially used to ”support services routines” [16] soon changed to replicate them. After that, Enterprise Architecture came to help in the service transformation and subsequently ”developing new converged services across different departments and functions.” [16].The orientation was to present flexible services and adopt the new strategies.
- People: Initially, people performed tasks utilizing technologies via digital literacy (skills to work and utilize digital platforms). The orientation was to free people from monotonous and repetitive tasks and focus more on creativity, empowering them.
- Governance: Governance went from using rules and policies via transactions and services to interactions and satisfaction. Later, with the orientation of opening people's participation and empowerment, the governance became co-creative and participative.

Also worth mentioning is the E-Government Maturity Models. Maturity models, more precisely e-government, are models that serve as a guide to ”assess or rank e-government portals. In order to assess electronic services provided to citizens, an appropriate e-government maturity model should be selected” [17]. E-government maturity models range from 4 to 6 stages:

- The first stage focus on the ”presence of the web”, which means having technology connected to the internet to communicate, share information, and so forth.
- The second stage understands the interaction between citizens and government through government portals and websites, improving the quality of information, enhancing it, and even allowing online transactions.
- The third stage is similar to the second stage, but also focused on interoperability, making the e-government portals a single-entry point.
- The fourth is also related to the two previous stages,giving attention to the personalization, where the web portal can and needs to be personalized according to the citizen's preferences or needs.
- The fifth and sixth stages are analogous and highlight the e-participation (citizens can participate in e-government applications like forums, votes, to name a few) and integration between government systems and e-portals, making their communication work smoothly.

These presented models are only two of the various related to digital transformation in governments. The goal was to provide a background for this dissertation, presenting a bit of history and a couple of theoretical models already created. It is also relevant to verify that some information presented in them will also be present in future sections of this dissertation, such as communication using a portal or the notion of customers' satisfaction.

Technologies that aid DT

Models, frameworks, and approaches to digital transformation in government are fundamental but irrelevant if they cannot be implemented. One way to apply them is to use technologies, specifically "emerging technologies". Technologies like Artificial Intelligence (AI), Robotics, and Internet of Things (IoT) have many applications in the most varied scenarios, and government is no exception. It is documented, for example, that blockchain can help secure e-government systems (Markusheuski et al.) [18] or that IoT can enhance it (Papadopoulou et al.) [19]. In this dissertation, the focus, as mentioned, will be on Process Automation technologies and systems.

2.1.2 Digital Process Automation

Digital Process Automation, as seen before and although less mentioned or studied than other technologies like AI (academically speaking), is highly relevant and vital to DT. Despite existing plenty of models and frameworks, only a few are related to government digitalization, and from those few, the majority are very high level and theoretical. In this subsection, the State-of-the-Art regarding Digital Process Automation is presented, mentioning terms as Business Process Management, Workflow Automation, Process Modeling, and Robotic Process Automation.

(Business) Process Management

Business Process Management is, from the OutSystems website, "the practise of designing, executing, monitoring, and optimizing business processes." [20]. "BPM is, in essence, a management idea. Organizations perform better when they pay explicit attention to their business processes from start to end ... To do this well, it is essential to understand the steps that are ... part of a business process, ... the people who are involved in these steps, the information that is being exchanged and processed ..., and the technologies that are invoked when executing the various steps. The optimization and alignment of all these elements will enable an organization to improve the generation of a particular product or service, for example by speeding up the business process in question, by making it more efficient..." [21]

In other words, it uses a set of activities or tasks, web services, processes, layouts, user input, and many others, with a specific goal. For example, the payment of a specific government service comprehends several steps such as filling and sending the necessary information by the citizen through

the government website, treatment of said information by the employee, usage of a notification service to notify the citizens of the status of their request, and so forth.

There are three types of Business Process Management [22]:

- Document-centric – create a document at the end of the process;
- Integration-centric – integration between this process and other company's software, such as processes or tools;
- Human-centric – business processes that require human intervention.

Regarding these types of BPM, their definition is not relevant, but they help understand more about them and their usability.

Workflow Automation

Workflow Automation is similar to Process Automation, even sometimes being used interchangeably. The main difference between them is that Workflow Automation "is the automation of the completion of specific tasks. Process automation is about a more holistic automation of a sequence of tasks" [23]. In other words, the focus of workflow is more on the necessary tasks, given a specific activity. Process automation or Business Process Management, on the other hand, focuses on various activities or components necessary to reach a specific end goal, while also analysing and monitoring them. As seen before, Business Process Management is a more complex and complete way to automatize processes and tasks in order to reach a goal. The idea is that both can be used in harmony, using technologies that will be mentioned in future subsections, to improve government processes (in this case) and enhance digital transformation.

Robotic Process Automation

Even though the objective of this dissertation is on Business Process Management (and Business Process Model (BPM) or workflow), it is worth knowing and understanding what Robotic Process Automation (RPA) means. RPA is also an "emerging technology" where a software agent or bots (the "robot") mimics the actions performed by the employee in order to reduce time and burden. The idea is to have an employee do some action, like sending an email which requires several steps such as searching for the citizen's email, and the bot recording those actions, later replicating what was done. In conjunction with Business Process Management, this technology can massively improve the process automation of organizations (and in this case, government) [24] but it is still in its infancy, scholarly research wise.

2.1.3 Low-Code Technology

Usually, even though AI, Blockchain, IoT, and many other technologies are very efficient and crucial to companies and government's digitalization, they raise many issues such as privacy concerns and environmental consequences [25]. Adding to those, the implementation of new technology can be very expensive and complex, requiring experts that might not be available. The idea of a low-code approach relates to a programming method that is done through visual elements and drag-and-drop instead of the conventional programming systems and languages, making the implementation faster and easier, among others. This last topic presents how it can be used in process automation and DT as a whole.

Low-code Approaches

Low-code, as mentioned before, is a "visual approach to software development, " [26] which means that "you can abstract and automate every step of the application lifecycle to streamline delivery of a variety of solutions" [26]. This means that programmers do not need to be highly specialized, which can be hard to find and costly, and the development of the application is usually faster, more interactive, and more agile. Due to the visual aspect, stakeholders can better understand what is being created and propose changes accordingly (instead of waiting for the "final stages of development" to change something) [26]. Low code can also be very useful for process automation for the same reasons mentioned before. Mentioning that, the focus of this dissertation is on using low-code in process automation, using the software provided by OpenText [27]. Is important to note that each government (or company responsible) should use the technology that better suits it.

OpenText

OpenText Corporation [27] is a company focused on the development of Enterprise Information Management (EIM) software. EIM software (or system) helps create, store, and manage information. This information is vital to the organizations that use them such as governments, that deal with large amounts of data ranging from citizens' information to other departments or organs. From the EIM solutions available, which are state of the art in the management of information, the one chosen is OpenText that supports Digital Process Automation and Dynamic Case Management – AppWorks [28], which is vital to the automation of processes by creating and using BPMs, lifecycles, web services and other components that will be explained further in this dissertation.

2.1.4 SLR 1: Current DT representations

The following subsections mentions the steps required to perform the Systematic Literature Review (SLR), Figure 2.2:

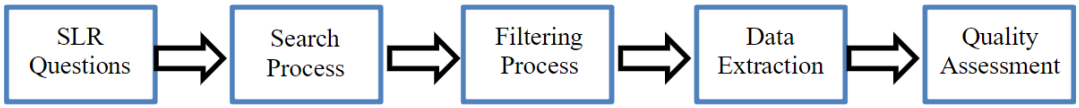

Figure 2.2: Steps required to perform the SLR

SLR Questions

Since the focus is to better understand the current representations (state of art, frameworks, and models) of Digital Transformation, with focus on the government entity and process automation, the SLR presented has the goal to answer the following questions:

1. What is the current state of art/frameworks/reference architectures used in digital transformation, preferably in government and related to process/workflow automation?
2. What are the main problems and difficulties regarding digital transformation and process/workflow automation in the government?
3. How are approaches and technologies (preference in low-code ones) used in government entities' digital transformation and process/workflow automation?

These SLR questions aim to better understand the main problems and difficulties in the topics mentioned and if technologies, approaches, or representation models are enough to aid in those difficulties.

Search Process

The figure bellow, Figure 2.3, demonstrates the steps used to gather the relevant records, using a similar approach to PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [29], although more simplified.

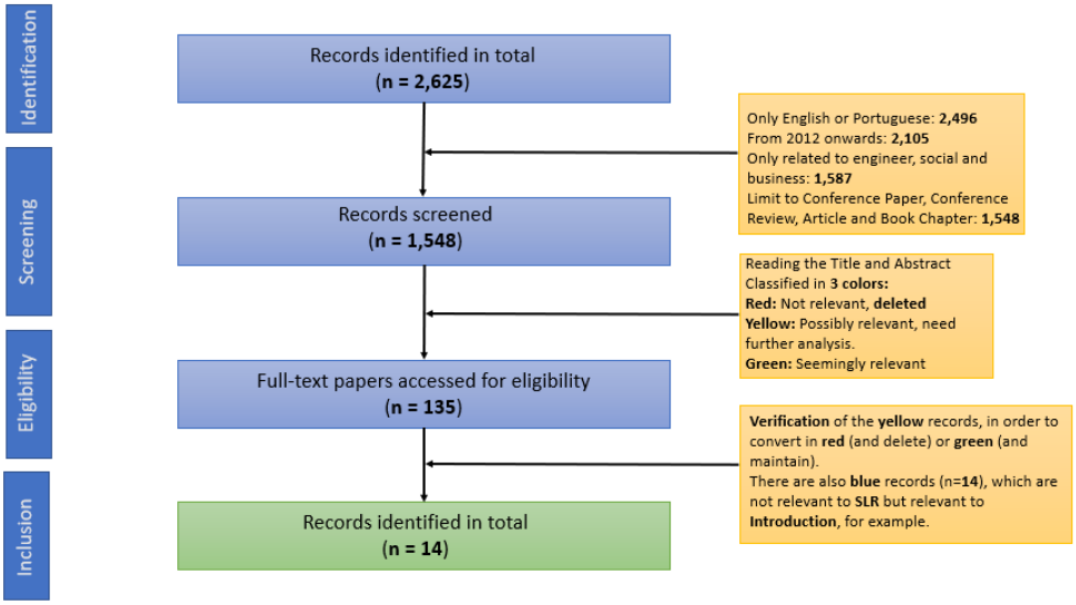

Figure 2.3: Model followed in selecting the relevant documents from the initial search

The SLR resorted to the Scopus database, with the query search, Figure 2.4:

(Model OR framework OR (reference AND architecture) OR architecture OR (state AND of AND art) OR (state-of-art)) AND (((process OR workflow) AND automation) OR (digital AND transformation) OR (application AND ((low AND code) OR low-code))) AND (government OR public)

Figure 2.4: Search Query

Regarding the query, the first part is about the first question, mentioning representation models used in DT. Terms that had more than one word (reference architecture, for example) were separated by the operator AND, meaning both should exist simultaneously. Between different terms, the operator OR was used to search by those terms individually, if necessary. All these terms were inside parentheses, followed by AND, meaning that at least one term needed to be used in the search. The following group of terms, Process OR Workflow, AND automation, Digital AND Transformation, and Applications AND low AND code OR low-code, followed the same rules as the previous one, with at least one necessary used in the search. These two groups were already enough to answer the questions proposed, but because the goal was to direct that search towards the government sector, a third group was created, containing Government OR Public. Public was mentioned because it is related to government, terms like Public Sector or Public Entity/Organization. This last group of terms is separated from the others with AND, meaning that the search query must contain one of the two, Government or Public, which is the specification of the questions proposed.In the first step, Identification, 2.625 records in total were identified. They had to be filtered by only English or Portuguese because English is the official language

and Portuguese is the native one, adding that it could give a better and more specific insight regarding Portuguese Government entities. This presented 2.496 records. Since Digital Transformation is a relatively new topic but quickly changing and improving, a year limitation was chosen, where only papers from 2010 onward (starting last decade) were selected, reducing their number to 2.105. Scopus also allows users to filter records based on topics. Because of that, only the ones related to engineering, social, and business were selected, discarding topics like health and entertainment (1.587) and limited to conference papers, conference reviews, articles, and book chapters (1.548). Since only one database was used in this search, no duplicate records were found (Scopus already filters them). After this phase, 1.548 were deemed relevant to the screening phase.

Inclusion-Exclusion Criteria

In this phase, the records were exported to an excel file (.cvs) and classified as red, yellow, or green, Figure 2.5.

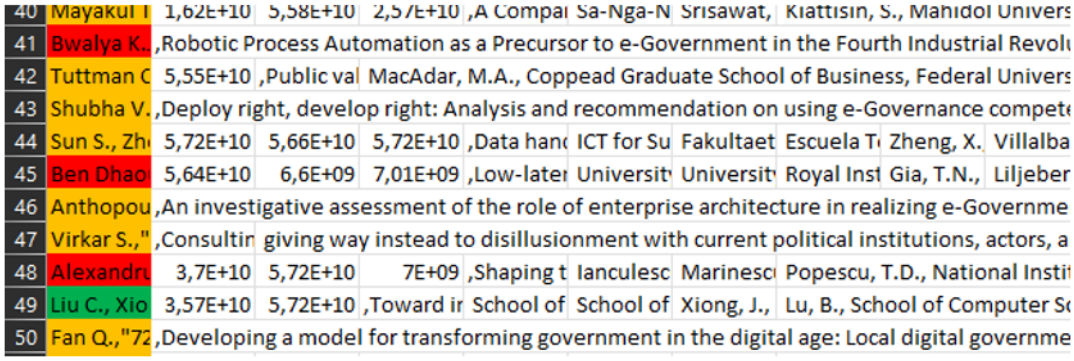

Figure 2.5: Section of Microsoft Excel related to the Inclusion-Exclusion Phase

The screening process was the following: first, the title reading, where records with irrelevant topics would receive a red mark and later be discarded. If the title was inconclusive of its relevance, a further inspection of the abstract was made, which marked records that were not related as red (later discarding them), with the assumed ones receiving a green mark. If its relevance was still unknown, a yellow mark was given. The remaining records from this step were 135, which proceeded to the next phase, Eligibility.In this phase, a full-text glance was performed to turn the yellow-marked records into green or red (following the same steps as the previously mentioned ones). After this final phase, 14 records were deemed relevant to answer the proposed questions. A new type of classification is also present here, blue records (14), which were not relevant to answer the proposed questions but could be helpful in other parts of this work, for example, in Introduction.

Data Extraction

The figure below, Figure 2.6, represents the distribution of the papers identified, from 2010 to 2021.

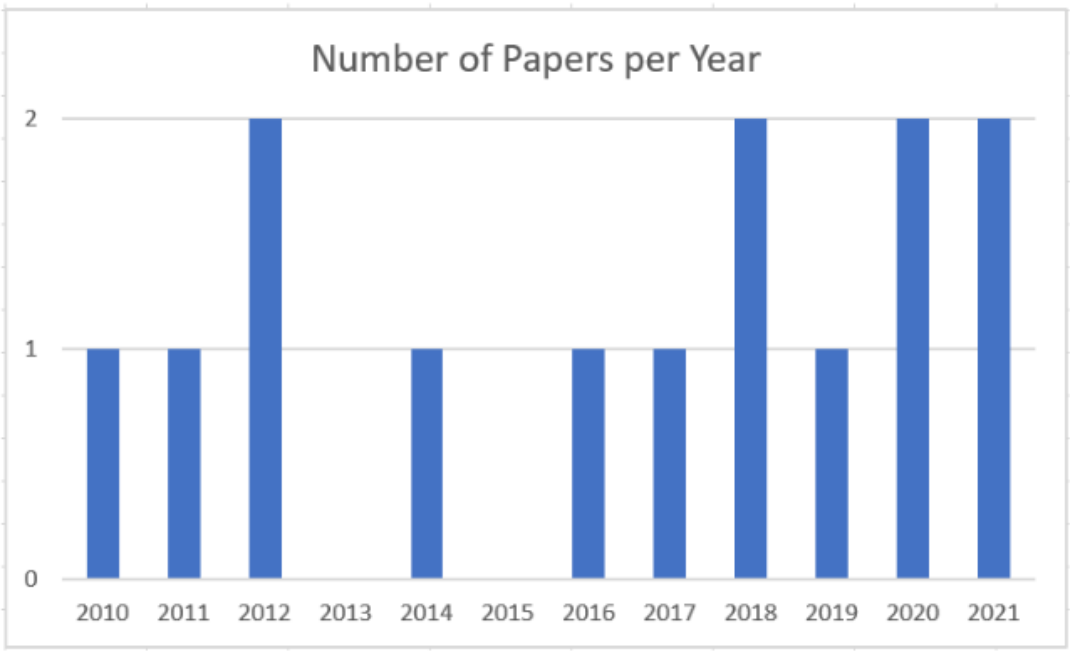

Figure 2.6: Number of records per year, from 2010 to 2021

From its observation, it is possible to infer that since the last five years, the number of records has almost doubled (from 5 to 9 presented). Given that the number of records might not be significant, it is relevant to note that this can lead to a biased conclusion. However, this can be simple to resolve by going back to Scopus and performing a search filter by date. The Figure 2.7 shows the number of created records each year.

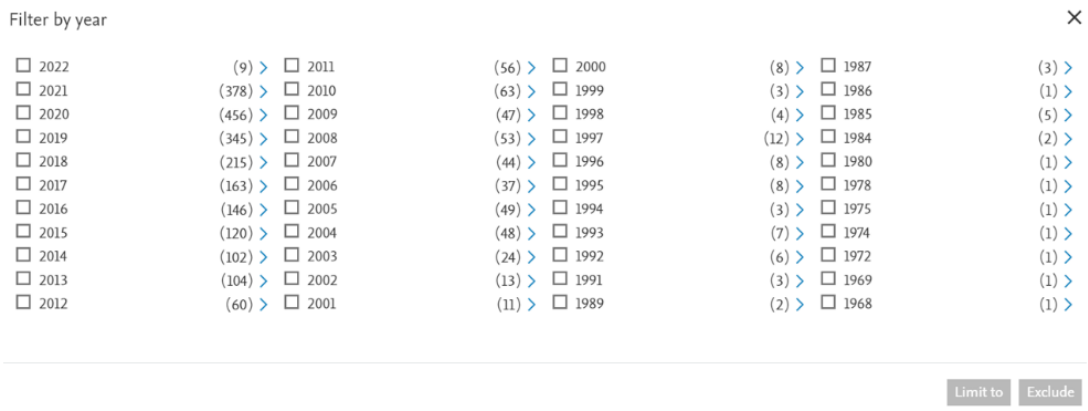

Figure 2.7: Print-screen from the number of records per year

Regarding the types of records obtained, 57% are conference papers from conference proceedings and book series,and 43%, articles from journals.
Two tables were created, with the first one, Figure A.1, containing general information like the papers' contribution or its findings, and the other one, Figure A.2, containing information more specific, used to answer the proposed questions. From the SLR performed, it is possible, without going into much detail since this is only for background and motivation behind this thesis (and the information is present in the tables created):

- Even though most of the papers containing the mentioned topics only appeared in recent years, there should exist some holistic/ general approach regarding the topics mentioned, but that is also specific enough for organizations to know what it involves. For example, most of the obtained papers or are too focused on their specific single country as with [2] (China) or [8] (Ghana). Also, or they present very abstract and not specific solutions such as [30], which only mentions the benefits and areas of impact, presenting a "conceptual framework for public enhancements" without going into much detail in how technologies to use/necessary modules, or they are very specific and without providing much information in other cases, as in [4] which only mentions the technology used, LOTUS DOMINO (now HCL Domino) [31] or presents workflows specific to their problem.

- Few organizations realize its true potential or do not know how to measure its adoption, even though these are rapidly becoming relevant ("real-world" and scholar wise), with very successful cases of its implementation. This is mentioned in [30] where "public organizations are often seen as the "dinosaurs of the digital era", lagging in their digital transformation".

- Organizations are afraid to adopt them because of the problems previously mentioned, adding to inexperience and need for training, [9] and [10], perceived risks like security, [2], costs [4], lack of flexibility [14], and interoperability issues, with relation with legacy technologies and methods [7], for example. Government is even more apprehensive because of the risks associated with large quantities of data (big data) that need to be stored and maintained, for example, from transactions between citizens and the government [26]. Adding to this is the necessity of reliable and advanced communication methods since government organs are complex entities, which takes time to implement/learn how to use, what might be perceived as not worthy. Also, the potential job losses raise concerns for the employees [10].

- Most of the papers obtained do not mention low-code, or it is unknown if the applications used contain some low-code elements. Most of the papers explain the technologies used with the majority of them mentioning "workflow", [9] and [32], and other terms as "automation" [10], mobile [7] or servers/databases [4].

Quality Assessment

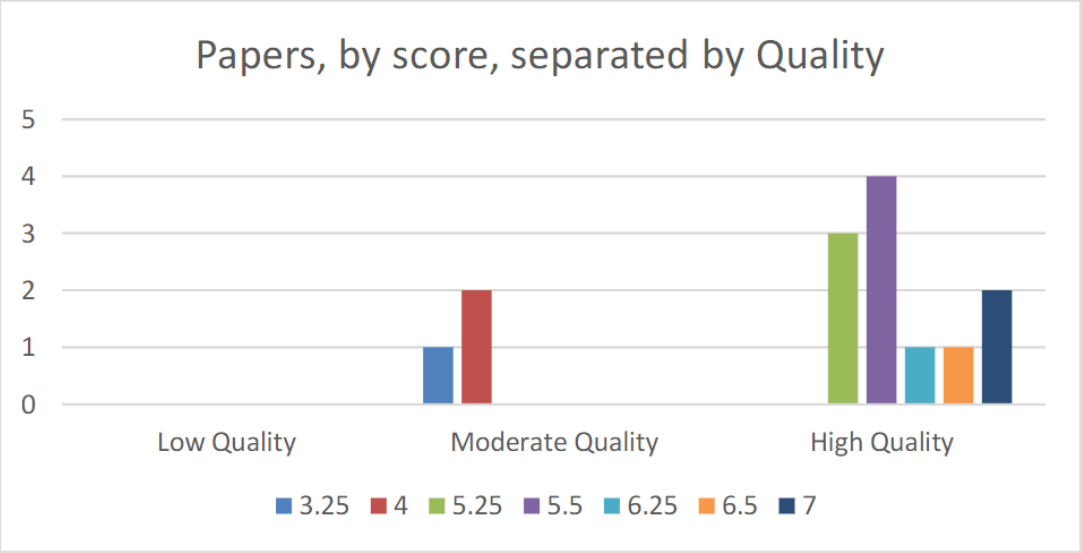

Figure 2.8: Number of Papers, grouped by scores (colors) and separated by Quality

In order to verify the quality of the presented reports, this subsection presents the quality assessment performed. Since this is a simplified SLR, as previously mentioned, tools like A MeaSurement Tool to Assess systematic Reviews (AMSTAR) [33] will not be used nor bias validation or reliability of the authors. The table in Figure A.3 presents the results and the followed rules. After these values, a second table, Figure A.4, is created, presenting the score per record and percentage of quality obtained. The median overall score for the obtained reports was 5.5, and the mean was 5.411 (with a population standard deviation of 1.059). 0 out of 14 were rated as low quality (0-2), 3 as moderate quality (2-5), and 11 as high quality (5-7). These values were taken from the AMSTAR score and adapted to this SLR, Figure 2.8.

2.2 State of the Art

2.2.1 SLR 2: Modules present in the DT of government processes

First, in order to find the general modules, a second SLR was performed. The decision to create a second SLR and not aggregate them together by a query (as it is usually done) is because they are different and try to answer different questions. The first SLR, as previously seen, is more towards the background and motivation of the presented work, with the second SLR more specific to this dissertation's contribution.

This SLR follows the same steps and methodologies as the previous one and,as such,only the dif-

ferences/principal aspects will be mentioned. The proposed question is the following:

- What are the main modules, used by Enterprise Information Management Systems, in the Digital Transformation of government's procedures/processes/functions?

The used query was the following, Figure 2.9:

((Enterprise AND Information AND Management) AND (Systems OR Solutions)) AND Government AND ((Digital AND Transformation) OR (Process OR Automation) OR Digitalization OR Low-Code OR (Low AND Code))

Figure 2.9: Search Query

Briefly explaining the query (since how it operates was explained in the previous section), the objective is to search for records that contain enterprise information management systems or solutions in government, associated with digital transformation, process automation, and low-code. It was not limited to the digitalization of government procedures because it would leave very few or even no records, and it is possible to retrieve essential modules/solutions for other digital transformation entities. For example, a system that deals with marriage and associated transactions [34] can have modules in common with procedure transactions.

Figure bellow, Figure 2.10, shows the steps taken regarding identification and filtration of the relevant papers:

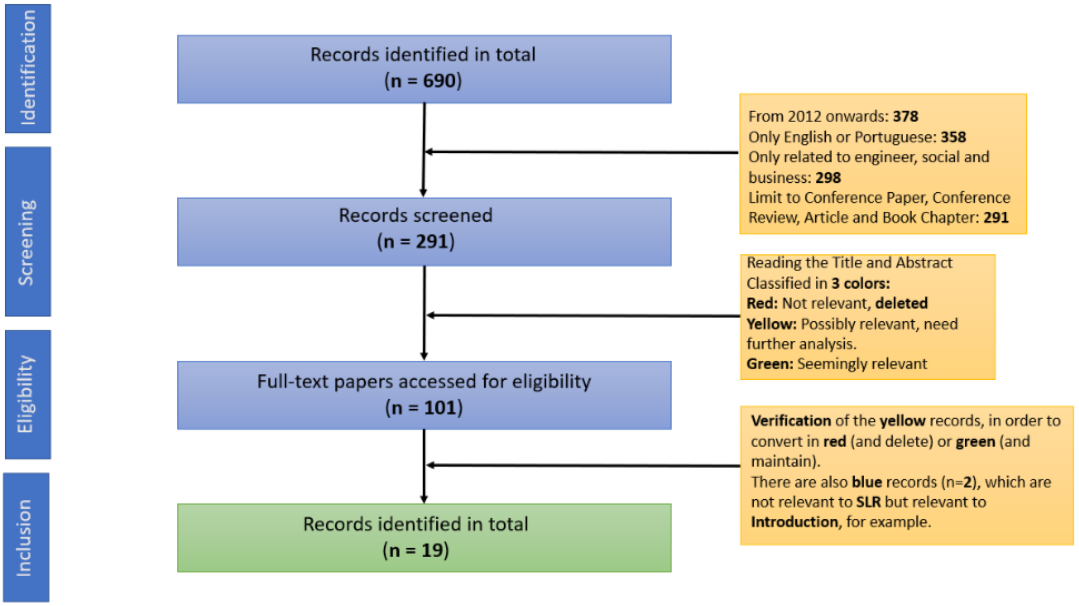

Figure 2.10: Model containing the steps taken when selecting the relevant documents

Data extraction

In the data extraction step, two tables were created, with the first one comprehending the models retrieved, Figure A.5, and the second, Figure A.6, presenting the same information but now mentioning the most common modules and if the papers obtained contain them.

The information obtained from the SLR will be presented in the following sub-section, State of the Art Topics, after the Quality Assessment.

Quality Assessment

The quality assessment in this SLR follows the same methods as the previous one with some differences. Since the papers obtained needed to contain modules, the evaluation method is not if/how well they answer the question (since they all do) but how many methods and technologies each paper contains, from the deemed most common/essential(scoring from 0 if not mentioning any of them and 9 if mentioning all). Figure A.7 shows each paper score. The median overall score for the obtained reports was 5, and the mean was 4.895 (with a population standard deviation of 1.293). 0 out of 19 were rated as low quality (0-2), 17 as moderate quality (3-5), and 2 as high quality (6-9). This information is present in Figure 2.11.

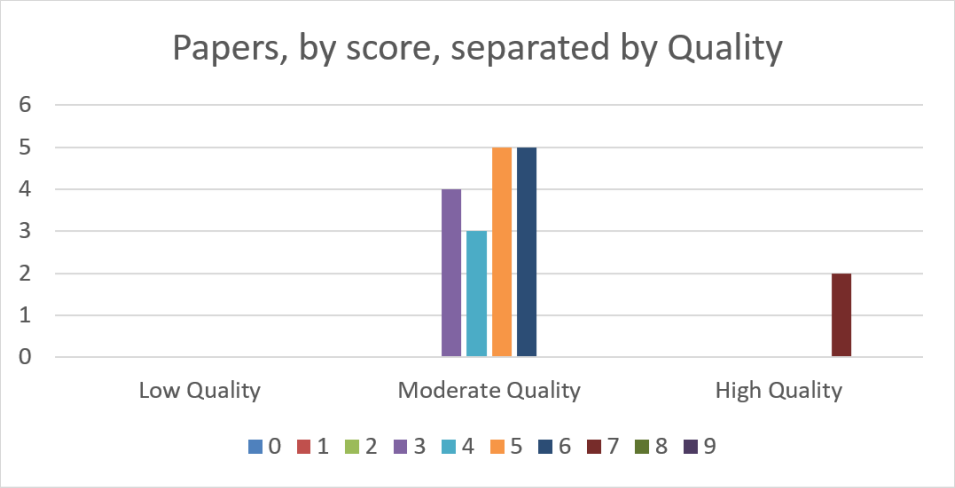

Figure 2.11: Number of Papers, grouped by scores (colors) and separated by Quality

As seen, most of the papers are in the moderate quality category, which means that the modules chosen are in fact the most common in papers of this type. There needs to be taken into consideration that most papers focus on few solutions and modules, for example [34] focus on marriage solutions and services, and that is the reason there are none that contain all (or even most) of the terms mentioned. This also means that having “low quality” or/and “moderate quality” papers is not exactly a bad thing.

2.2.2 Main Modules

The majority of the papers, as shown in the Figure A.5, contain the modules user portal or web pages, documentation, communication, (Web) Services (such as payment service), and a citizen application. From that figure, it was created (as mentioned before) a second figure, Figure A.6, presenting the most common modules found and if the papers contain them.

Explaining better the terms present in the second picture, referring to the papers that mention them:

- Business Process Models: Will be explained further in this dissertation, but are models that contain the graphical implementation of the existing business processes. [35] mentions different ”views” such as functional ”activity, sub-process”, dynamic ”control flow, information flow”, organizational ”actor, resource”, content ”product, artifact”, quantitative ”times, costs...”, and time-oriented ”version, variant”, also mentioning that the models are not final and should be improved ”the business process model is analyzed according to its structure, behavior, etc. and based on the findings the model is redesigned accordingly”. This paper also mentions that the Business Process Model and Notation (BPMN) is classified as a Dynamic process modeling language. [36] also refers that the Business process management modules are used to ”Automate and manage all aspects of business”.

- Web Pages: Comprise the interaction between customers and services. Each web page should have different security/permissions to restrict each employee to their specific task. The majority of papers contain this module, such as [34] that mentions an ” eGovernance application software that is accessible through a portal by citizens” or [37] that explains that the stakeholders interact with the National portal through the internet and that portal connects to the service providers (Ministries/Departments). Another common trait is that most of them mention the citizen’s interaction/web pages and not the employee’s, which this dissertation will focus on.

- Entities: Will be explained later but essentially are blocks that contain information. In this case, procedures, citizens, and others are all entities that will need to be represented in the models created. Some entities also contain a lifecycle, which needs to also be created. [32] mentions that ”connection types between function and entity are ... based on CRUDL functions: ”uses, views, creates, changes, reads, deletes, lists.” These actions or functions are vital when resolving procedures and each actor should perform the functions mentioned, such as reading a procedure or updating one.

- Documentation: Every document created or uploaded needs to be stored in a system, which needs to be reliable and secure. Usually, documentation is associated with an entity but can also be stored independently. [32] mentions the creation of an ”automatically generate user requirements

document". This automation is important and should be incorporated into the lifecycle of the process entity. There is also information that is not related to entities, such as web components, that should be stored in a different directory.

- Communication: Communication is not only between citizens or employees and the organization's application. Internal components and modules such as entities, sometimes require communication with other entities or modules. For example, the procedures entity needs to communicate with the notifications entity when citizens are notified that a procedure is completed. [34] mentions the prioritization of information and communication technologies in order to address "the ever increasing paper based manual systems and fostering digital approaches ...". An example of communication technology is the web service.

- Web Services: Regarding the web services, some papers such as [38] mention a term named Service-Oriented Architecture (SOA), which refers to the "ability ... to connect multiple services between diversity systems".In other words, is the usage of service interfaces that can be used and reused in other components of the application. In order to realize SOA, the best way is to use web services, which "are software systems designed to support interoperable machine-to-machine interaction over a network" [39]. These web services are vital in BPMN and low-coding approaches (mentioned further in this section). Several services perform some tasks, such as sending emails or completing payment, as cited by [34] that mentions a payment service done "through e-payment receipts portal".

- Security/Permissions: Each block of information, such as entities or web pages, requires security/permissions to limit the actions performed by the users that use them (such as limiting certain users to modify specific entities). This is directly correlated with roles and mentioned in [32], where different roles perform different functions.

- Roles: Few papers mention roles but they are very important especially if related to the security/permissions module. Roles related to security by grouping users and giving them group permissions instead of individual ones only. [32] mentions several sentences/templates in which different types of roles (Role N1 or Role M1) execute different functions (from the ones mentioned earlier). For example "Role N shall carry out; ... Role M shall approve ..."

- Citizen Application: As seen, the majority of the academic research regarding the topics treated here relates to the citizen application, such as seen in the [34] or [37]. Even if the focus in this dissertation is on the employee's side, it is also important to mention the citizen's, since the employee application needs to communicate with it to function properly. This can be achieved, for example, through web services, mentioned before.

EIMSs

After mentioning the leading solutions and modules necessary to create the reference architecture, the next step is to find what Enterprise Information Management System to use. From a search on the internet, more precisely in the website Gartner Magic Quadrant and Critical Capabilities [40], which allows the search and identification of the best-positioned technology providers, using two metrics, "Ability to Execute" (AE) and "Completeness of Vision" (CV) and dividing the providers into four parts: Challenger (>50% AE, <50% CS), Leaders (>50% AE, >50% CV), Niche Players (<50% AE, <50% CS) and Visionaries (<50% AE, >50% CV). After searching for the best providers of the previously mentioned modules, the following table was created, figure 9. It is important to note that not all providers are presented, for example, Adobe, but the intention was only to show that many EIMS are available and contain the exact solutions, meaning that each organization must choose what best suits them. It is important to note that not all EIMS solutions are presented, for example, OpenText also contains AI and Analytics which are not mentioned here, but the idea was to present solutions and to verify if most providers had them.

As mentioned before, the focus of this dissertation is on OpenText, advising again that each government (or the company chosen by it) should select what system suits them best, adapting the holistic view of the information presented here to their specific case.

Low-code Components

It is important to identify if the major low-code solutions to the state of art are the components used in low-code solutions, which should be present in most low-code solutions in the following figure 10.

	BPM (Business Process Models)	Entities	Collaborative Workspace	User Interface/ Homepages	Lifecycles	Security	Roles	Web Services	Organizations	Mobile Support
OpenText AppWorks	x	x	x	x	x	x	x	x	x	x
OutSystems	x	x	x	x	x	x	x	x	x	x
Oracle BPM	x	x	x	x	x	x	x	x	Unknown	x
IBM BPM	x	x	x	x	x	x	x	x	Unknown	x
SAP BPM	x	x	Unknown if collaborative	x	x	x	x	x	x	x

Figure 2.12: Table Showing what standard low-code components each technology solution contains

It is possible to notice that almost, if not, all the technologies that use low code have the same

components, supporting what was mentioned regarding the choice of what technology to use. In this dissertation's case, since the company in charge works with OpenText, that is the one chosen, OpenText AppWorks. The components will be explained in future sections.

Having identified the main components and modules from the State of the Art performed, the following section aims to identify and explain the most common use cases.

3

Use Cases

Contents

This chapter presents the goals and basis of the use cases used to treat the government procedures, also presenting the explanation regarding each one.

3.1 Goals and Requirements

Depending on the goals that each government has, the way to resolve and deal with its processes can vary. Because of this, and since there are multiple use cases possible, only will be presented, in this chapter, the main ones: Resolve a Procedure, Reclassify a procedure, Create a Request, and Send a Notification Manually. These use cases were deemed most important because of the SLR previously mentioned and because of the real-world implementation used as a basis for this dissertation.

The use case Resolve a Procedure is the main one, that appears the most in the SLR performed, such as [34] when performing and resolving the marriage certificate. The goal is to resolve a procedure, even if they have different types (by trying to make it in a generalized way).

The second use case is the Reclassify a Procedure and it is important because sometimes the procedures are not created correctly or classified correctly and instead of having to create new ones, having the ability to reclassify them makes that interaction easier and faster. This use case comes from a real-world application and because it was one of many use cases there, it was deemed relevant to present here.

The third use case is Create a Request. This is also a use case that comes from the real-world implementation but is also found in the SLR although not directly (such as using the communication module mentioned, for example, by the [34] when the system communicates with the citizen). Sometimes the procedure requires more information, for example from the citizen or a specific employee. In that case, the employee in charge needs to be able to send that request to the specific person.

The last use case, Send a Notification by Hand, comprises those cases in which the citizen wants to receive the notifications by hand instead of electronic methods such as email or SMS. In this case, the employee needs to send the notification through mail or a carrier and wait for the reception of it, to proceed with the procedure's resolution. This case comes from the real-world implementation and was deemed very important because it aids people that normally are not so much knowledgeable about technology such as elderly people and because it incorporates the use case to notify a citizen, one of the major modules identified either in the SLR or in real-world scenarios.

There are plenty of other use cases possible but these were deemed not only very important but also easy to generalize and present. Because of that, this research and solution proposition only considered these use cases (and their ramifications, such as the Cancellation of a Procedure that is included in the Resolve a Procedure).

This chapter is divided into 4 sections, with which one containing the use cases mentioned. Each

section comprises the use case model and the tables explaining it (summary, basic course of events, and others). Since most use cases have common parts (such as beginning activities with the log-in), those parts will not be presented in further appearances in order to not repeat information. Also relevant is that since these use cases were retrieved from the specific real-world government implementation, other governments may have different (or more/less) use cases that are not presented here. Given this, it is they should adapt what is presented here to their specific cases.

3.2 Use Case Diagram 1: Resolve a Procedure

In this first use case, Figure 3.1, there are two actors, the Revision Group Employee (as the name infers, it is the employee that belongs to the Revision Group) and the Resolution Group Employee (similarly). The goal here is to resolve a procedure. To achieve it, first, the employee (Revision) needs to fill in the necessary information and send it to another employee (Resolution). This employee will then decide if the procedure's resolution is positive or negative, completing the human tasks, being then finalized through several automatic activities (such as generating documentation related to it).

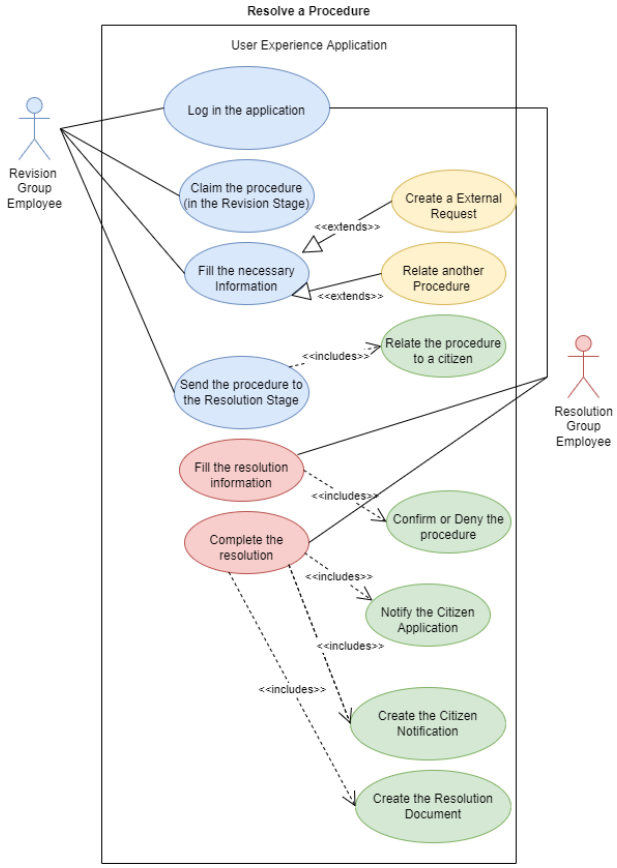

Figure 3.1: Resolve a Procedure

The first step, Table 3.1, is to log in the platform (such as AppWorks Experience [28]) using the credentials.

Table 3.1: Activity 1-1: Log in in the Application

Name	A1-1: Log in in the Application
Summary	The user logs in to the application using the respective credentials.
Actors	Revision Group Employee and Resolution Group Employee.
Preconditions	The user needs to be created in the Experience application (Directory Services one, responsible for the log-ins).
Basic Course of Events	1. The user accesses the application URL. 2. The user inputs the login credentials. 3. The user confirms the credentials and presses the login button.
Alternative Paths	1. In step 2, if the user forgets the credentials, there is an option to recover them. In this case, the application takes the user to another page where the credentials lost can be retrieved, by providing specific personal information.
Exception Paths	1. In step 2, if the user inputs the wrong credentials, the system will ask for them again suspending temporarily the access if a set number of attempts, depending on the application, are performed.
Post-conditions	The user, successfully, logs in the application and enters the personal page.

In the second step, Table 3.2, the employee claims the procedure from the procedures list.

Table 3.2: Activity 1-2: Claim the procedure

Name	A1-2: Claim the procedure
Summary	The user claims the procedure to himself.
Actors	Revision group employee.
Preconditions	The procedure must be created before, in order to appear in the tasks' inbox.
Basic Course of Events	1. The user accesses the tasks' homepage. 2. The user accesses the tasks' inbox. 3. The user selects and opens a procedure. 4. The user claims the procedure by selecting the action bar button Claim.
Alternative Paths	1. There is a possibility that the procedure was assigned to the employee either by another employee or by the system. In those cases, the employee does not need to claim the procedure.
Exception Paths	None
Post-conditions	The user is now the owner of the procedure.

After, in Table 3.3, the employee verifies and fills in the necessary information to send the procedure to the next state.

Table 3.3: Activity 1-3: Fill the necessary information

Name	A1-3: Fill the necessary information
Summary	The user fills in the necessary information to send to the next state.
Actors	Revision group employee.
Preconditions	The user must have claimed the procedure and it needs to be in the Revision state.
Basic Course of Events	1. The user navigates to a tab in the procedure. 2. The user fills in the necessary information there. 3. Repeat all the necessary tabs.
Alternative Paths	1. In all the steps, it is possible to revoke the procedure. 2. In all the steps, it is possible to send the procedure to another state, for example, reclassification (UC2) or misclassified. 3. In all the steps it is possible to fill in additional information such as creating requests (UC3) or relating the procedure to another.
Exception Paths	1. In step 2, it is possible to encounter errors when inputting information, such as when creating a request.
Post-conditions	The procedure is now ready to send to the Resolution state.

In Table 3.4, after the previous use case, the employee sends the procedure to the next state.

Table 3.4: Activity 1-4: Send the procedure to the Resolution state

Name	A1-4: Send the procedure to the Resolution state
Summary	The user sends the procedure to the next state, resolution, after filling in the necessary information.
Actors	Revision group employee.
Preconditions	The procedure must have the required information filled.
Basic Course of Events	1. The user navigates to the action bars. 2. The user presses the button “Change States” and then “Go to Resolution”. 3. A pop-up opens and the user confirms the action.
Alternative Paths	1. In the pop-up notification, it is possible to deny the operation and return to the Revision stage, where changes can be made.
Exception Paths	1. If an error occurs a popup message will show, mentioning where the error is or to contact the administrator of the system.
Post-conditions	The procedure is now in the Resolution state.

The employee, Table 3.5, fills in the necessary information to resolve the procedure.

Table 3.5: Activity 1-5: Fill in the resolution information

Name	A1-5: Fill the resolution information
Summary	The user completes the information necessary to resolve a procedure.
Actors	Resolution group employee.
Preconditions	The procedure must be in the Resolution state.
Basic Course of Events	1. The user claims the procedure (A1-2). 2. The user navigates to the resolution tab. 3. The user decides if the procedure’s resolution is positive or negative. 4. The user fills in the remaining information, such as notes of the procedure’s resolution.
Alternative Paths	1. In all the steps, the user can choose to send the procedure back to the Revision state. 2. In all the steps, the user can revoke the procedure.
Exception Paths	1. If the required information is not filled in, a red notice appears, reminding the user to fill it in. 2. If the information is wrong, such as the wrong format, the same red notice appears mentioning the error.
Post-conditions	The procedure is now ready to be completed.

In the Table 3.6, following the previous use case, the employee completes the resolution.

Table 3.6: Activity 1-6: Complete the resolution

Name	A1-6: Complete the resolution
Summary	Having the procedure decision and information filled, the user now completes the resolution.
Actors	Resolution group employee.
Preconditions	The procedure must be in the Resolution state and its resolution must be filled.
Basic Course of Events	1. The user navigates to the action bars. 2. The user presses the button Change States and then Complete Resolution. 3. A pop-up opens and the user confirms the action.
Alternative Paths	1. In the first step, the user can choose to revoke the procedure. 2. In the pop-up notification, it is possible to deny the operation and return to the Resolution state, where further changes can be made.
Exception Paths	1. If an error occurs a popup message will show, mentioning where the error is or to contact the administrator of the system.
Post-conditions	The procedure is now completed.

3.3 Use Case Diagram 2: Reclassify a Procedure

In this use case, Figure 3.2, the only actor, the Revision Group Employee, wants to reclassify a procedure that was incorrectly created/classified. In this case, the procedure needs to go first to the Reclassify state before going to the Resolution state. In the Reclassify state, the procedure will be changed in order to correct the error identified, such as changing its department. (Important to note that, as mentioned before, repeated activities will not be presented in the activity tables but will appear in the model).

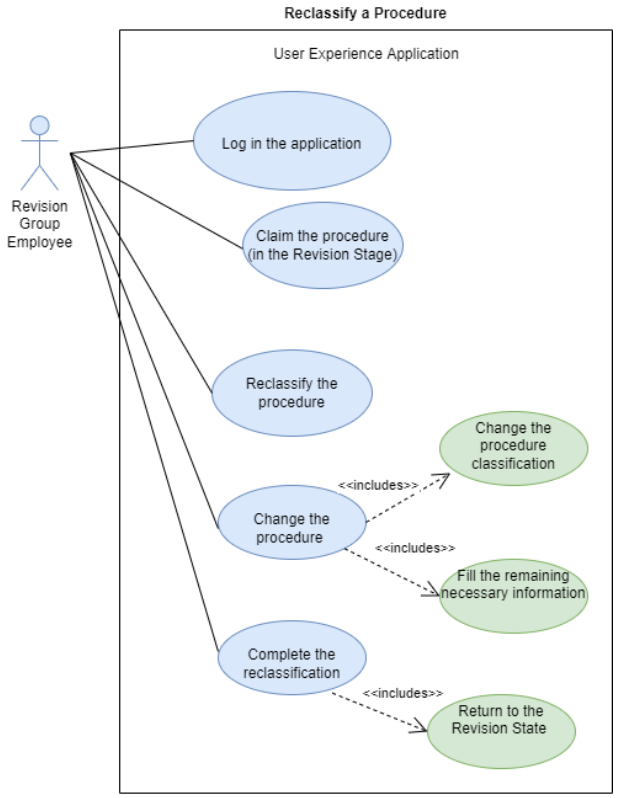

Figure 3.2: Reclassify a Procedure

In the Table 3.7, the user sends the procedure to the reclassify state.

Table 3.7: Activity 2-3: Reclassify the procedure

Name	A2-3: Reclassify the procedure
Summary	The user sends the procedure to the reclassify state.
Actors	Resolution group employee.
Preconditions	The procedure must be in the Revision state.
Basic Course of Events	1. The user navigates to the action bars. 2. The user presses the button Change States and then Go to Reclassify. 3. A pop-up opens and the user confirms the action.
Alternative Paths	1. In the pop-up notification, it is possible to deny the operation and return to the revision tab, where further changes can be made.
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service).
Post-conditions	The procedure is now in the reclassify state.

After that, the employee makes the necessary changes to the procedure Table 3.8.

Table 3.8: Activity 2-4: Change the procedure

Name	A2-4: Change the procedure
Summary	The user changes/reclassifies the procedure.
Actors	Resolution group employee.
Preconditions	The procedure must be in the reclassify state.
Basic Course of Events	1. The user navigates to the reclassification tab. 2. The user decides the new procedure classification. 3. The user fills in the remaining information, such as notes of the procedure's reclassification.
Alternative Paths	1. In all steps, the user can choose to revoke the procedure. 2. In all the steps the user can choose to send the procedure back to the Revision state.
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service).
Post-conditions	The procedure is now ready to be reclassified.

Following that, in the Table 3.9, the employee sends the procedure back to the Revision state, now reclassified.

Table 3.9: Activity 2-5: Complete the reclassification

Name	A2-5: Complete the reclassification
Summary	Having filed the necessary information, the procedure is now ready to be reclassified.
Actors	Resolution group employee.
Preconditions	The procedure must be in the Reclassify state and its reclassification must be filed.
Basic Course of Events	1. The user navigates to the action bars. 2. The user presses the button Complete Reclassification. 3. A pop-up opens and the user confirms the action.
Alternative Paths	1. In the pop-up notification, it is possible to deny the operation and return to the reclassification tab, where more changes can be made. 2. It is possible to return to the Revision, without performing the reclassification.
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service).
Post-conditions	The procedure is now reclassified.

3.4 Use Case Diagram 3: Create a Request

The goal of the third use case, Figure 3.3, is for the actor here, Revision Group Employee, to create a request regarding missing information/documentation. Requests can be internal, external, and citizen (the standard ones, more can be added if necessary). The first type, internal, is directed to an employee or a group inside the government (or employee application). External requests refer to external entities (outside the government) through the usage of e-mails, for example. The last type, citizen, refers to the citizen that is related to the procedure, and the communication is done through the citizen's application.

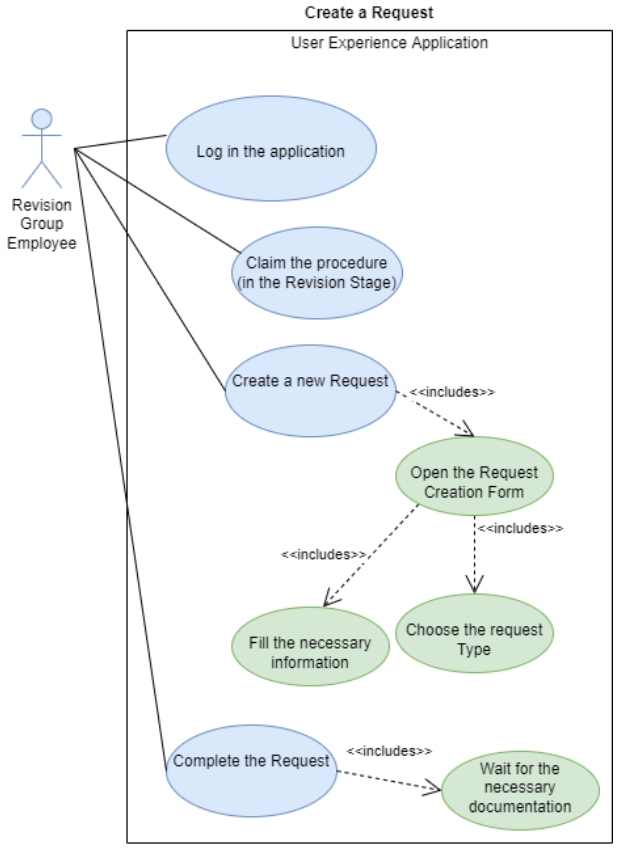

Figure 3.3: Create a Request

The employee, when necessary, creates a new request, Table 3.10, choosing its type and filling in the necessary information.

Table 3.10: Activity 3-3: Create a new Request

Name	A3-3: Create a new Request
Summary	The user can create a new request for the procedure.
Actors	Revision group employee.
Preconditions	The procedure must be in the Revision state.
Basic Course of Events	1. The user navigates to the Requests tab. 2. The user creates a new request in the list. 3. A pop-up opens and the user chooses between the 3 types of requests (Internal, External, or Citizen). 4. The user chooses the type of request and fills in the necessary information regarding the type chosen. 4.1. If the request is internal, the user needs to choose which internal group/user to request. 4.2. If the request is external, the user needs to choose which external entity to request. 4.3. If the request is a citizen, the user sends the request to the related citizen. 5. Confirm the creation of the request in the button Confirm.
Alternative Paths	1. In all the steps, the user can cancel the operation and return to the requests tab. 2. In step 5, the user can search in the requests list for the request created and open it, seeing its information.
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service). 2. It is only possible to have one citizen request open at any given time. If the user wants to create another citizen request it first must complete the previous one or an error message will appear. (This limitation does not happen with the other type of requests).
Post-conditions	The procedure's request is now created.

In the Table 3.11, the employee completes the request upon receiving the necessary documentation.

Table 3.11: Activity 3-4: Complete the Request

Name	A3-4: Complete the Request
Summary	The user completes/closes the request previously created.
Actors	Revision group employee.
Preconditions	The request must be created and in either the Pending Internal, Pending External, or Pending Citizen states.
Basic Course of Events	1. The user navigates to the home page, Requests inbox. 2. The user opens the specific request. 3. Upon receiving the necessary documentation (visible in the documentation Tab), the user navigates to the action bar. 4. The user presses the button Complete Request. 5. A pop-up opens and the user confirms the action.
Alternative Paths	1. In step 5, the user can cancel the operation and return to the requests tab.
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service).
Post-conditions	The procedure's request is completed.

3.5 Use Case Diagram 4: Send a Notification by Hand

In this last use case, Figure 3.4, there are three actors, the Revision Group Employee, the Resolution Group Employee, and a new one called Notification Group Employee. This use case refers to the act of sending notifications by hand. The standard way for citizens to receive notifications is by SMS or email. There are cases, however, that citizens (mostly elderly or technology-illiterate people) prefer to receive them by hand. In this case, the Notification Group Employee must confirm the dispatch of the notification and the confirmation of its reception. It is important to note that this process is only performed after the procedure is completed/resolved, which is done by the previous actors, Revision Group Employee and Resolution Group Employee, in the use case Resolve a Procedure.

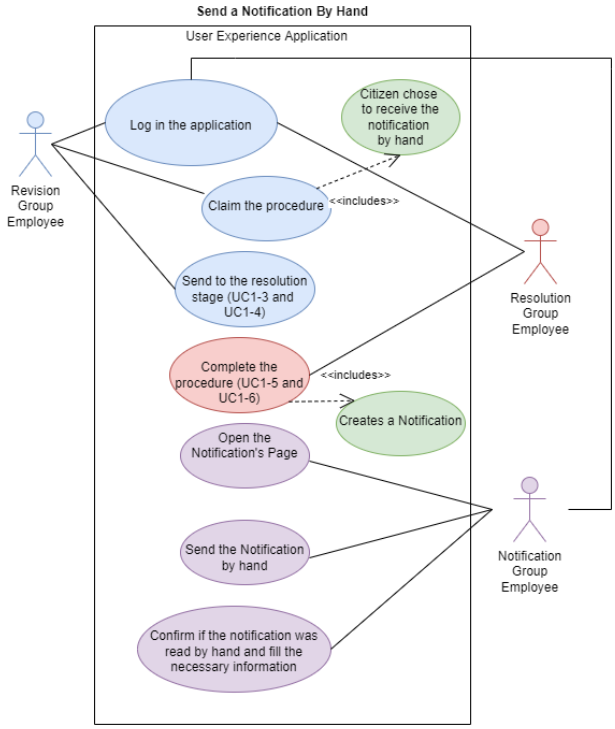

Figure 3.4: Send a Notification by Hand

In the Table 3.12, the employee responsible for the notifications opens up the notification created automatically from a completed procedure.

Table 3.12: Activity 4-5: Open the notifications page

Name	A4-5: Open the notifications page
Summary	The user searches and navigates to the notifications page.
Actors	Notification group employee.
Preconditions	The procedure must be completed/resolved, the notification created and the citizen must have checked the receive notification by hand option.
Basic Course of Events	1. The user navigates to the left side which contains all the possible home pages. 2. The user finds the “Notifications Tasks” page. 3. The user opens the page in step 2. 4. The user finds, in the notifications list, the specific notification of the procedure created. 5. The user selects the notification and presses the button “Open” in the action bar.
Alternative Paths	1. In step 4, the user can search the notification in the list by sorting various properties such as created date or the related citizen’s name. 2. In step 5, the user can also open the notification by double-clicking in it.
Exception Paths	1. If the notification does not appear in the list is most likely because of an error that occurred and, in that case, an error message would have to appear when resolving the procedure.
Post-conditions	The user found the notification and is inside its tasks page.

In the Table 3.13, the employee performs the sending of the notification by hand.

Table 3.13: Activity 4-6: Send the notification by hand

Name	A4-6: Send the notification by hand
Summary	The user sends the notification to the citizen by hand.
Actors	Notification group employee.
Preconditions	The procedure must be completed/resolved, the notification created and the citizen must have checked the ”receive notification by hand” option.
Basic Course of Events	1. The user navigates to the action bars 2. The user is notified that the notification is already sent to the citizen (by hand). 3. The user presses the button Sent Notification by Hand. 4. A pop-up opens and the user confirms the action.
Alternative Paths	1. In the pop-up notification, it is possible to deny the operation and return to the Notification tab
Exception Paths	1. If an error occurs, for example regarding a web service or wrong information provided, a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service).
Post-conditions	The procedure is now in the Waiting for Read Confirmation state.

Upon the confirmation that the notification was received and read, the employee fills in the remaining necessary information, Table 3.14.

Table 3.14: Activity 4-7: Confirm if the notification was read by hand and fill in the necessary information

Name	A4-7: Confirm if the notification was read by hand and fill in the necessary information
Summary	The user confirms if the notification was read by hand and fills in the remaining necessary information.
Actors	Notification group employee.
Preconditions	The procedure must be in the Waiting for Read Confirmation state and the notification read by the citizen.
Basic Course of Events	1. The user is notified that the citizen received and read the notification. 2. The user, in the Notifications tab, verifies that the obligatory read property is checked. 3. The user verifies if the additional information is automatically filled in by the citizen application. 4. The document regarding the evidence of notification is created, automatically, by the system.
Alternative Paths	1. If X days (decided by the government entity) pass and the confirmation of reading the notification is not received, the creation of the notification's evidence starts automatically. 2. In steps 2 and 3 the user can add more information to the evidence notification if deemed necessary. 3. In steps 2 and 3, in very specific cases, the user can input the necessary information manually instead of waiting for it to be updated by the citizen's application.
Exception Paths	1. If an error occurs, for example regarding missing the obligatory provided information (read check, for example), a popup message will show, mentioning where the error is (or to contact the administrator of the system if the error is on the application side, for example in a web service). 2. If the necessary information is not filled in (either automatically by the citizen's application or manually by the notification group employee), the creation of the document regarding the notification's evidence is not started. 3. If an error occurs internally with the application, for example regarding a web service, a popup message will show, asking the user to contact the administrator of the system for further details.
Post-conditions	The life-cycle of the notification is now finished and the evidence notification document is created.

4

Solution Proposition

Contents

This chapter presents the solution proposition based on the use cases mentioned earlier, the real-world DT implementation (which, as referred to, served as a basis for this dissertation), and the state-of-the-art presented in the chapter Background.

This solution proposition consists of three parts. The first part presents the domain model regarding the main identified entities and their lifecycles, present in each project created. The name project was given to each module such as Procedures or Notifications, that contains the entities and other components, because it is the name that the low-code application used to implement the real-world scenario utilizes.

The second part consists of the creation of a reference architecture containing the projects (modules), their components, and their connections/dependencies.

The last part is the application implementation which explains how to implement the information presented in the previous two sub-sections in a real-world scenario using low-code technologies.

The goal is to incentivize governments to perform or improve their DT, namely of its processes, resorting to methods that can ease it such as low-code or EIM solutions.

4.1 Domain Model and Lifecycles

This section is divided into two. The first one, which contains the domain model, Figure 4.1, in the form of a Unified Modeling Language (UML) model, explains the various entities from the major projects identified, with also their relationships. Despite been present the dependencies between projects, these will only be mentioned in the second section of this chapter. The second sub-section presents the lifecycle of the entities that contain it, detailing the different states and their transitions.

4.1.1 Domain Model

There are four main projects present in the domain model, which are the Common Parts, Procedures, Notifications, and External Entities. These will be detailed further in the reference architecture, but for now, the focus is only on their entities and how they relate.

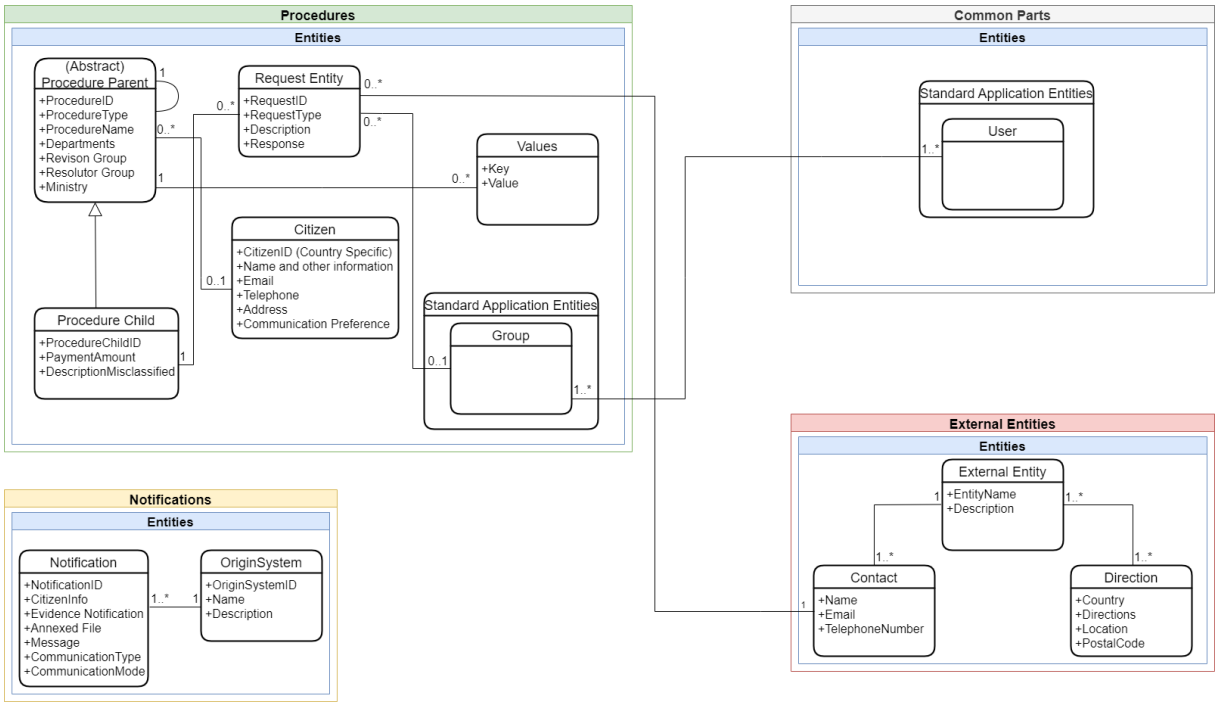

Figure 4.1: Domain Model

The Procedures Project contains the entity Procedure Parent, which is the main entity, abstract, and refers to the government procedures that the employees deal with. This entity contains properties such as Procedure ID, Procedure Type (different procedure child), Departments, Groups, and others. Important to refer that all the properties in this and the other entities will not be presented because they might depend on the government's application/case or are internal, used in the programming component - such as a Boolean flag. This entity can be related to another of the same type (in this model it can be related to only one but this should be changed if necessary). Since a Procedure Parent is abstract, there need to be procedures that realize it. In this model there is only one mentioned, the Procedure Child, in order to avoid presenting very specific information, leaving to each government how to define the Procedure Child and how many they are. This Procedure Child inherits from the parent the properties, previously mentioned, and adds a few new ones such as Procedure Child ID, the Description, and the Payment Amount. Further mentions of Procedure Child or Parent will be mentioned as a whole, called Procedure, to simplify the information presented.

Procedures are also related to only one citizen, or none if they are created in the employees' application and still do not have a citizen assigned. Entity Citizen contains the Citizen ID, the Name and other personal information, the Communication Methods (SMS, email, address...), and the Communication Preference (by hand, by SMS...). Citizens can have multiple procedures or none (if they still have not created one).

Procedures can also have zero or more values. Entity Values are properties that are not created pre-

viously in the Procedure entity but are useful to the procedure in question. For example, if an employee wants to add a new property, specifically related to a procedure, which will appear in the documentation created when completing it (such as a note mentioning that the procedure had to be revised 5 times), the employee will have to create a new value with the property key being ”notes” and the value property being ”Had to be revised 5 times”.

A Procedure is also related to one or more request entities (mentioned earlier) which contain the properties Request ID, Request Type (if it is Internal, External, Citizen, or even other - not presented here), Description, and Response. Each Request entity only relates to one Procedure.

Lastly in this project, there are the Standard Application Entities, which are entities not created by the developers but that already exist in the application, internally, if the necessary libraries are present. Examples of them are Group and User. Here, one relevant is the Group entity that allows the employees to be grouped into groups, such as Reader or Modifier, allowing them to have specific permissions and authorizations (especially useful when closing a procedure, in which the permissions of all the users that can access that procedure are converted into read-only, not allowing further changes to it).

In the project Common Parts, which contains the parts common to each project, there is only one entity, a Standard Application Entity, called User (the one mentioned before). This entity relates to one or more Group entities from the Procedures project. This means that each user (employee) is inserted into one or more groups and each group contains zero or more users.

In the Notifications project, there are two entities, Notification, and OriginSystem. OriginSystem is relevant because it allows the identification of which system created the notification, such as the employee’s application, the citizen’s application, or even an external one. This entity contains the Origin System ID, the Name, and the Description. There is also the notification entity, the main entity here, that refers to the notifications created. This entity is related to only one originSystem (with each OriginSystem containing zero or more notifications) and contains the properties relevant to the notifications such as Notification ID, Communication Type (Notification or Communication - a simple notice, and so on), Mode (SMS, email,...), the Message, Attached File, and others.

Lastly, there is the External Entities project, which contains the entities External Entity, Contact, and Direction. The External Entity contains the properties Name and Description, one or more Contact entities, and one or more Direction entities. The Direction entity contains the properties referent to a direction, such as Country, Postal Code, or Location. Directions are only related to one External Entity (if no External Entity contains that location, that direction should be deleted in order to save database storage space. This, however, can change if the government identifies the necessity that a Direction can exist while not being related to an External Entity),

The Contact entity contains the Name, the Email, and the Telephone Number (and other potential contact properties). Contact is also related to the request entity from the Procedures project mentioned

earlier. Each Request entity contains one Contact Entity which relates to only one External Entity (each External Entity can have various Contact entities, such as various telephone numbers, but each Contact belongs to only one External Entity). This is similar to Directions where if the government deems it necessary, Contacts can exist even if not related to an External Entity. For security reasons, Request entities should have only access to the contact information of External Entities. This, however, can change depending on the government's project preferences/guidelines).

4.1.2 Entities' Lifecycles

This sub-section presents the lifecycles of the only entities identified that have them: Procedure, Notification, and Request. The transitions between states can be manual (user icon), conditional (white paper icon), and automatic (green check icon). Automatic transitions occur when all the activities in the state are completed, manual transitions through an action button, and conditional when a condition is met. Note that user and automatic transitions can also have conditions.

Important to refer, once again, that each government should make the necessary changes to the information presented, such as adding more lifecycles. Another thing is that they are modeled as they appear in AppWorks and not as a UML standard or other type, present for example in the [41], because of two reasons. First, it makes it easier to migrate from the model to the application AppWorks, since it is basically the same (improving time and understanding), while also being easier to understand for people not familiar with either AppWorks or UML. Second, this type of model contains more information, such as states or activities, which are harder to represent if modeled as a UML standard, for example.

4.1.2.A Procedure's Lifecycle

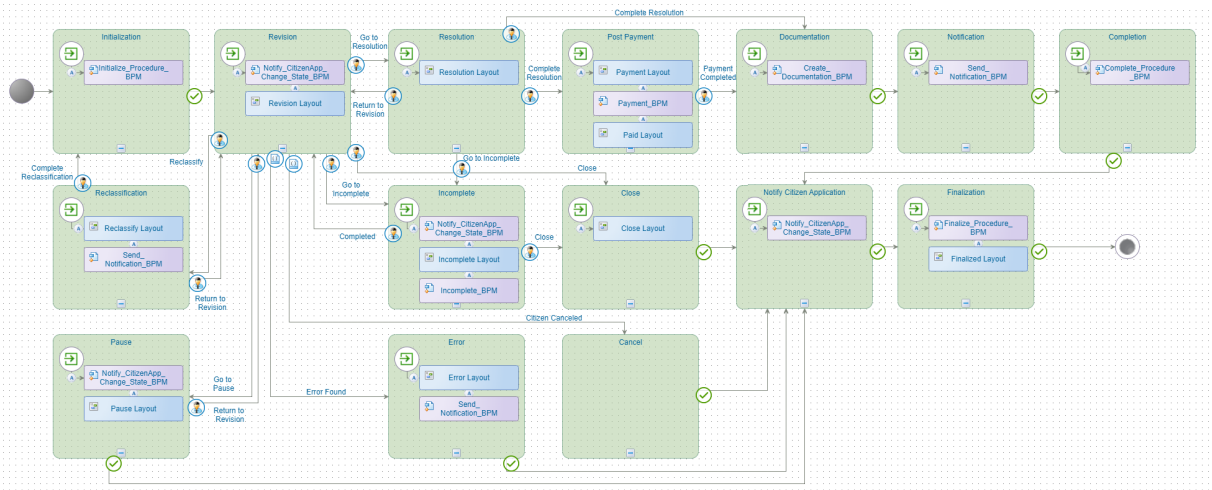

Figure 4.2: Procedure's Lifecycle Model

The Procedures entity starts in the Initialization state, which is where the steps to initialize a procedure are done. It contains the retrieval of information from the citizen's application, the confirmation that the procedure was paid, attributes the read/write employee's permissions relative to the type of procedure, and others. From this state the procedure goes to the Revision state, where the Revision Group Employee verifies the procedure's information, adding or modifying it if necessary. It is also communicated to the citizen's application that the procedure is in this state through the usage of the Notify_CitizenApp_Change-State_BPM. From the Revision state, the procedure can go to various states: the main path state, Resolution, and the alternatives, Reclassification, Error, Incomplete, Cancel, Close, and Pause states.

The Reclassification state is where the employee, after identifying that the procedure is incorrectly classified (wrong department/type/others) is sent to be modified. After this state, the procedure goes back to the Initialization, performing the same validation checks as before. It can also return to the Revision, not performing the changes.

The Error state is where the procedure goes if a critical error is found. This state first opens the layout which allows the employee to describe the error, sending then a notification to the citizen reporting the error and that the procedure will be closed. After that, it goes to the Notify Citizen Application state, which is where the employee application (internally) communicates the procedure's state with the citizen. This state is the previous one before reaching the Notify Citizen Application state, which in this case is the Error state. After that, the procedure proceeds to its finalization in the following state, Finalization.

There is also the Incomplete state, where the procedure is sent if the referred information is not enough to send it to its resolution and cannot be inputted by the employee. From this state, the procedure goes back to the Revision, now with the documentation/information necessary, or goes to the Close state, where the procedure is closed and terminated (similar to the Error and Cancel states) and sent to the Notify Citizen Application state. The procedure can also go directly to the Close state from Revision if the employee chooses to (with the action bar Close).

Cancel is also another possible state, and is similar to the Close state but differs in whom chooses to close the procedure. Close state is when it is closed, either manually by the employee or automatically from the Incomplete state. Cancel, however, is when the citizen, through the citizen's application, decides to cancel it.

Another possible state is Pause, where the employee or the system (automatically) sends the procedure to a ”frozen” state. This is especially useful with resolution claim-timed procedures or, for example, when the revision employee started the revision process but will not be able to finish it anytime soon. From this state or the procedure goes back to Revision or finishes and goes to the Notify Citizen Application state.

The following main path state is the Resolution State, where an employee from the Resolution Group

decides if the procedure has a positive or negative resolution. From this state, the procedure can go to the Post Payment state, the Documentation state, the Incomplete state (mentioned before), or even return to the Revision state.

The Post Payment state is an optional state where procedures require another payment (after the initial payment and the creation of the procedure). Here, the citizen will be notified of the payment value, leaving the procedure waiting for its confirmation. Important to note that the following main path states are automatic and should not require employee input.

The next state is Documentation, where the resolution document and other relevant documents are created in the Content Server application, linked to this application/procedure. If the documentation is confirmed to be signed by the competent entities (through BPMs and web services), the procedure continues to the next state. If not (error or faulty resolution) then the procedure returns to the resolution state.

After this state, the procedure goes to the Completion State, where the Close_Procedure_BPM, as the name implies, performs the completion of the procedure (in the Citizens' application), communicating with the citizen application that the procedure was completed correctly or incorrectly, and performing the last property updates such as date completion.

The following state is the already mentioned Notify Citizen Application with the last one being the Finalization state that is common to all states and finishes the procedure in the Employees application, removing the ability to write in the related partition/folder inside the documentation/content server application (converting all permissions to read-only). After this state, the Procedure lifecycle, Figure 4.2, is now finished.

4.1.2.B Notification's Lifecycle

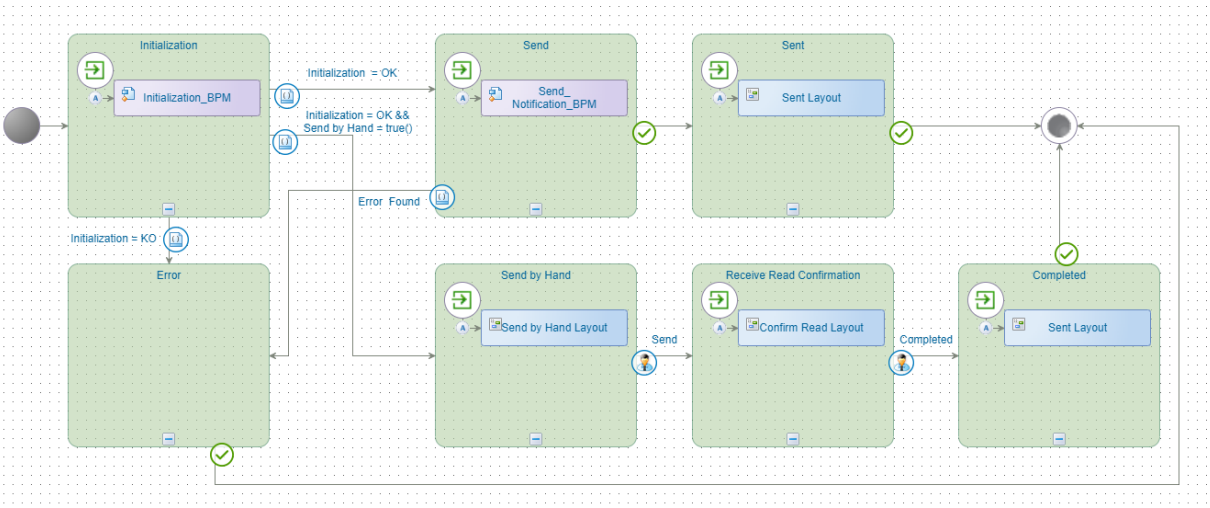

Figure 4.3: Notification's Lifecycle Model

The Notification's lifecycle, Figure 4.3, consists in the Initialization state, which functions similarly to the Initialization one in the Procedure's lifecycle. From here, the notification can go to the following state:

- Send (main path), if the initialization was performed correctly;
- Error, if an error occurred;
- Send by Hand, if no error occurred but the citizen chose to receive the notification in person.

From Send, where the sending of the citizen's notification is performed, it then goes to Sent, if no error occurred, finishing the lifecycle. This main path is fully automatic but it is possible, if necessary, to introduce human actions in the Send state, for example adding additional information or documentation.

If it goes to the Error state (which is also a possible state from the Send state, if an error while sending occurred), the notification aborts and finishes, not being sent to the citizen.

Send by Hand is the specific case in which a notification is sent manually, requiring a read confirmation. Here, the Notification Group employee revises and confirms the sending of the notification (Use Case 4), going then to the Receive Read Confirmation, where upon receiving it, the final values are filled, such as the read date. The last state is the Completed, where the notification is confirmed sent, read, and completed.

4.1.2.C Request's Lifecycle

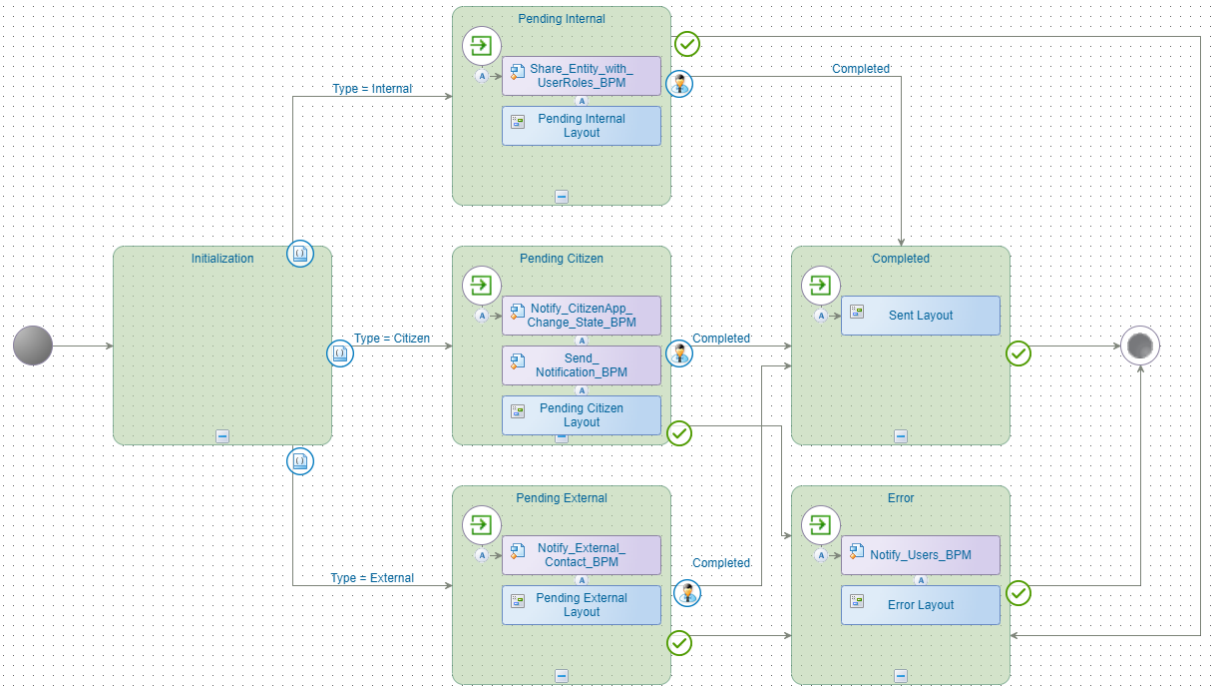

Figure 4.4: Request's Lifecycle Model

The last lifecycle, Figure 4.4, relates to the Request entity. This lifecycle starts with the Initialization state but differently from the other lifecycles there is no web service or layout here. This is only the first state which performs the verification of the request type, branching then into three alternative paths:

- Pending Citizen state, if the request is a citizen request;
- Pending External state, if the type of request is external to the government (employees) application;
- Pending Internal state, if it is internal.

All three states and their paths function similarly. Each state contains BPMs and web services that will perform the sending of a request to a citizen, an external entity, or an internal one. Pending Citizen, specifically, communicates with the citizen regarding the request, and with the citizen's application about the new state (similarly to the Notify Citizen Application state from the Procedure's lifecycle).

From these states, the request goes to the Error state, if some error occurred or goes to the Completed one if everything went as expected - both finishing the lifecycle.

4.2 Reference Architecture

This section presents the main projects' modules and the relations between them, through the creation of a reference architecture, Figure 4.5. It aims to provide a generalized view of the different modules in order to ease the transition from model to program/application. It is important to refer that each government should make the necessary adaptations to this reference architecture.

This reference architecture contains 5 projects:

- Common Parts
- Procedures
- Notifications
- External Entities
- (External) Citizen Application

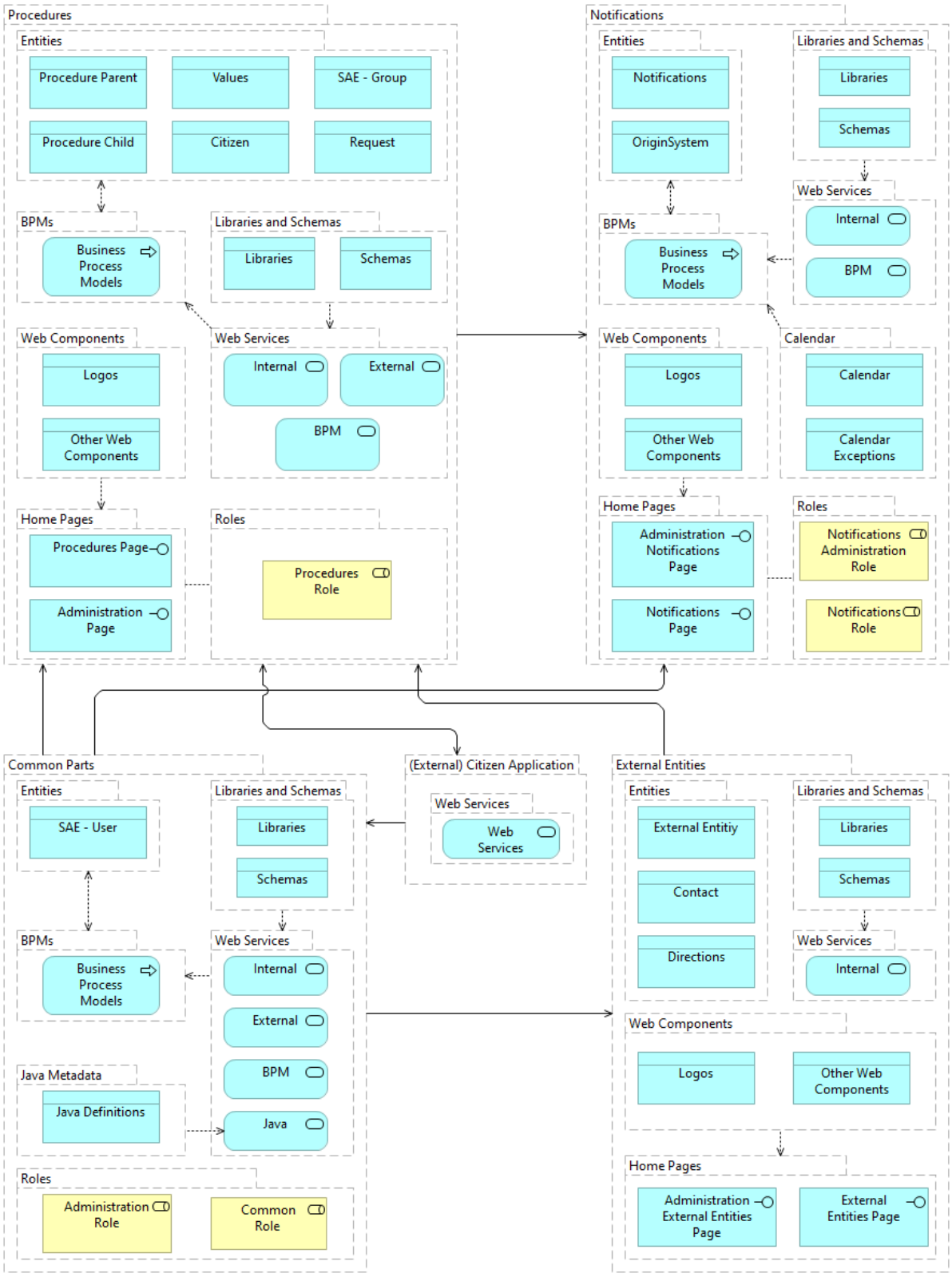

Figure 4.5: Reference Architecture

4.2.1 Common Parts

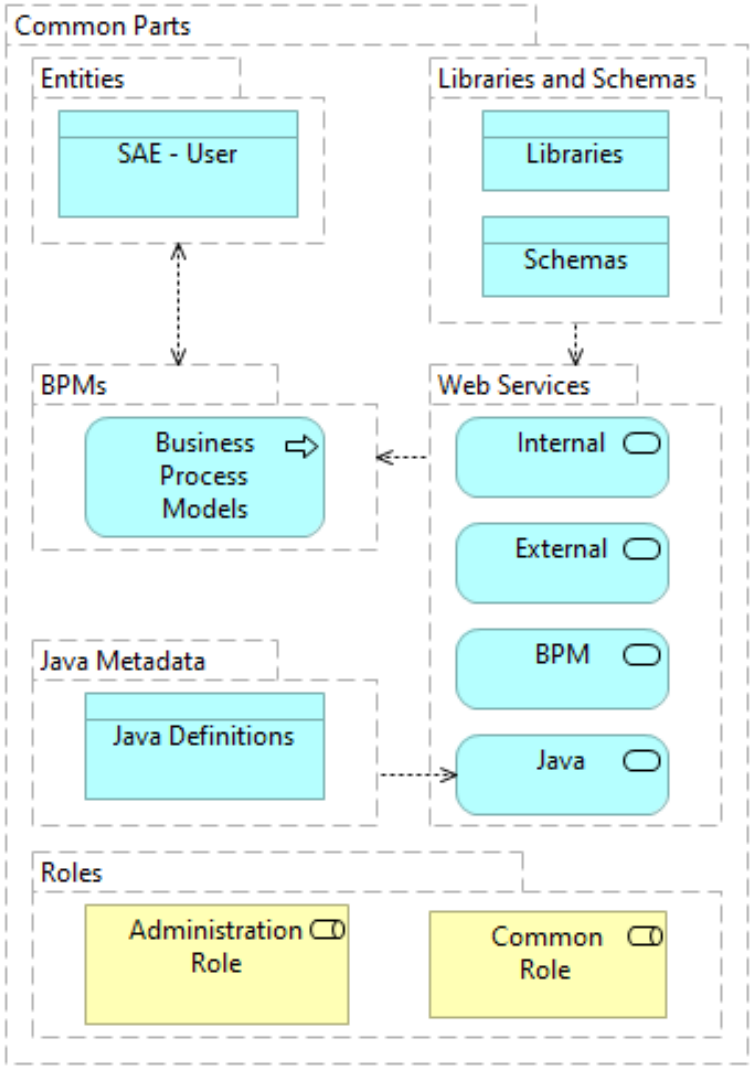

Figure 4.6: Common Parts Module

The first project, Common Parts, Figure 4.6, was created to contain all the parts (web services, business process models...) that are common to the various projects in a single location. The reason for this is the avoidance of circular dependency errors. Circular dependencies are detected whenever two objects reference each other [42]. An example of this is a project A, serving project B, with B also serving A. Because of this, A requires that B is compiled first but B requires A to be compiled first since both depend on each other and thus originate a circular dependency. One way to avoid this problem is to create another project containing all the parts used by A and B, making them not dependent on each other but on that common project.

This project contains a module/directory Entities, with the entity User, presented in the previous

chapter.

It also contains the Business Process Modelling and Notation (further mentioned as Business Process Models, BPM, since it is the name given by the low-code application used), which are a ”visual modeling language for business analysis applications and specifying enterprise process workflows” [43]. This means that it contains the models used to implement the government's business processes. The BPMs in this project are mostly related to common parts such as roles, users, or groups (also relating to the Directory Services application).

The BPM uses the module Web Services (that will be explained later) which uses the module Libraries, ”collections of prewritten code that users can use to optimize tasks” [44] and Schemas, ”a formal description of the structure or organization of a particular database” [45]. Both can be internal if they are pre-defined by the application or external if required and the application chosen allows it. Regarding web services, AppWorks specifically allows the architectural style Representational State Transfer (REST) and the communication protocol Simple Object Access Protocol (SOAP) [46]. The majority of them are SOAP (XML, with a body, envelope...), with the internal ones (predefined in the entity) being REST (use JSON [47]...).

Another module is the Java Metadata, which contains java definitions. Every (web) service that is created externally in Java (or another language) is stored here, which can then be used in other modules or projects. Normally, these web services are created because the application chosen to develop the DT does not have them or similar functionalities (for example, when integrating with other external applications such as the citizen's one).

There is also the Roles module that contains the created roles, referring to the employees' groups. In this project, Common Parts, there are two, the Administration Role which receives more permissions (administrative), and Common Role, more specialized (basic operations such as viewing homepages).

Lastly, there is the Web Services module that uses the Libraries and Schemas from the application and also the Java Metadata created. The four types are:

- Internal: Created by the application, such as predefined in an entity (read entity, update entity,...);
- External: Retrieved from other applications such as citizen's (get documentation,...);
- Java: Created from the Java Metadata module;
- BPM: From a BPM, it is possible to generate a web service, allowing its usage in other projects or applications (such as citizen's one). This can be specific to AppWorks.

A module that is not present is the module Schedules, which is where BPMs, or web services, execute at a specific moment in time, for example, every day at 6 pm. This module, however, was not created because it might be too specific to the application used, type of DT and because they can always be inserted in the Calendar Module (maybe in a subdirectory Schedules).

4.2.2 Procedures

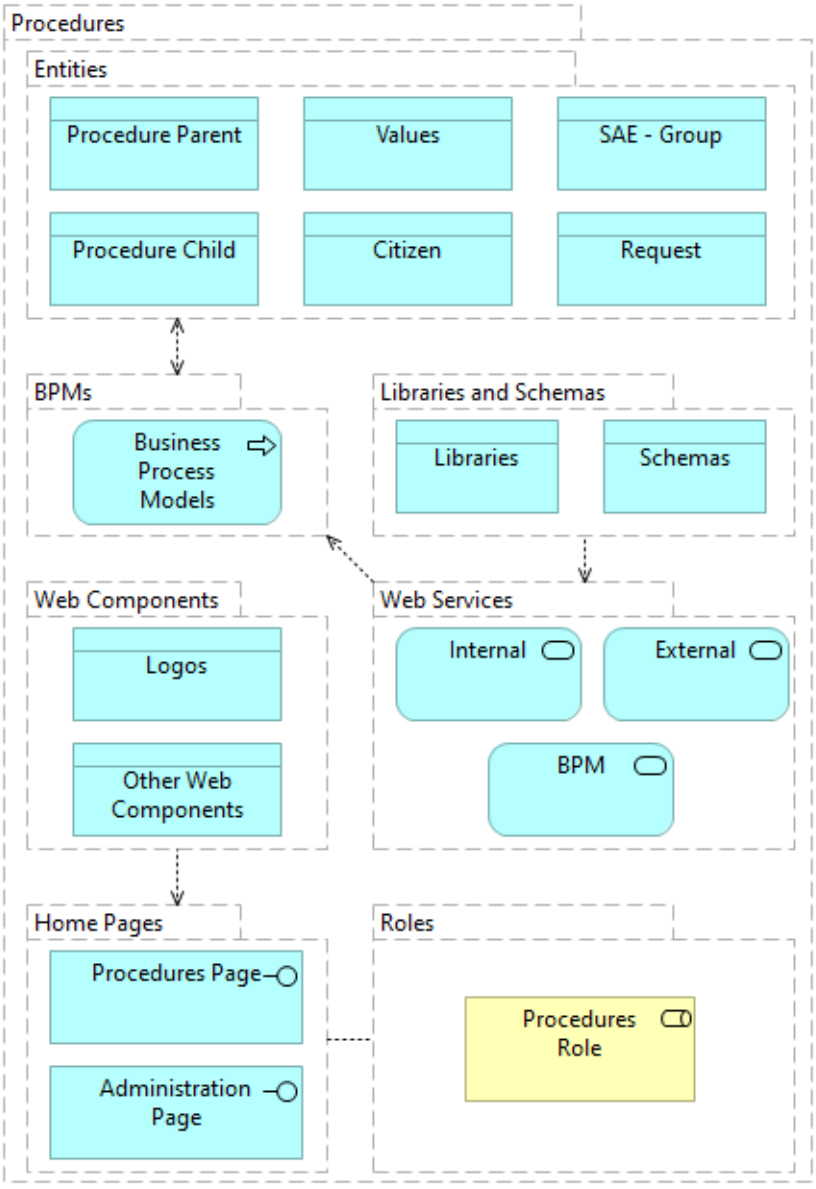

Figure 4.7: Procedures Module

The second project is Procedures, Figure 4.7, which uses the Common Parts, presenting similar modules. It also depends on the External Entities project, which will be mentioned later.

Procedures contain the Entities, already mentioned, and the BPM which are now specific to this project (such as creating a procedure or closing one).

This project also contains the Libraries and Schemas used by the Web Services module. Here, the Java web services are missing because only the Common Parts project contains them.

There is also the roles module which presents a new role, Procedure Role, specific to this project, giving new permissions regarding procedures such as creation or ability to read them.

Finally, and not present in Common Parts, are the Web Components and Home Pages modules.

Home pages are the front end of the application where the employees can create and resolve procedures. There can exist multiple home pages such as Administration Page, reserved only for administrative tasks (Administrative Role), and the Procedures Page, accessed by Procedures Role users. These home pages can use web components which are, for example, images with the government logotype.

4.2.3 External Entities

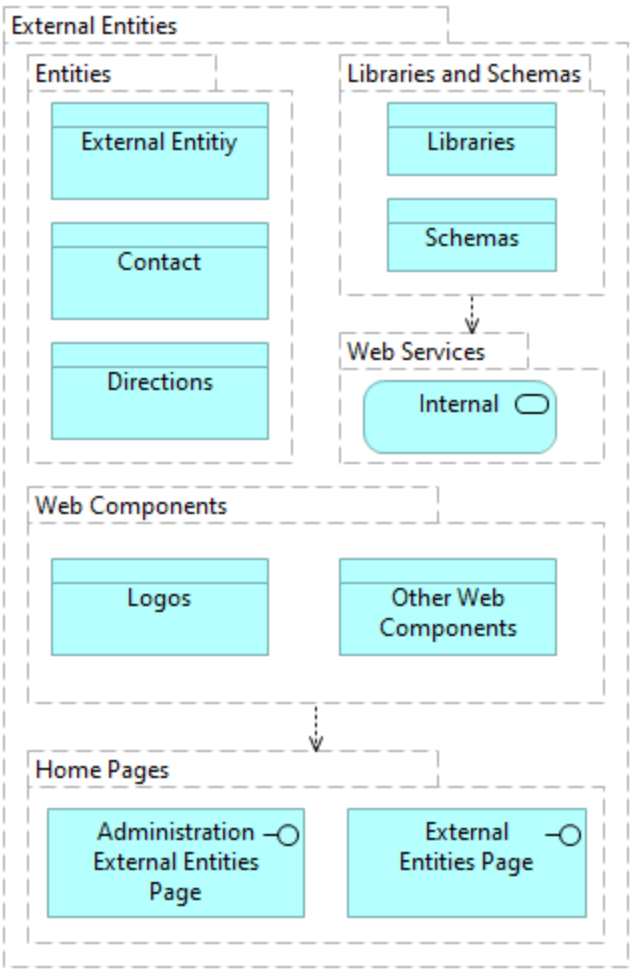

Figure 4.8: External Entities Module

Another project is the External Entities, Figure 4.8, related to the creation of external requests from the Procedures. It depends on the Common parts, containing Entities (already mentioned), Web Services (Internal), Libraries and Schemas, and the Home Pages, Administration and External Entities Pages, which contains the external entities' information and possible actions (such as creating or updating one).

4.2.4 Notifications

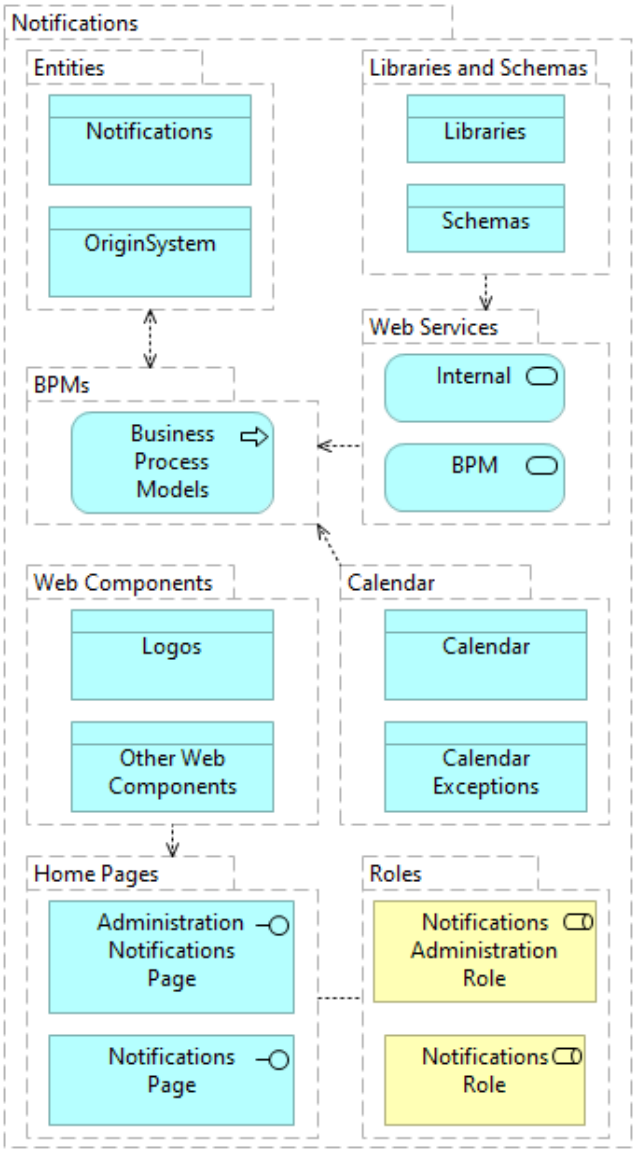

Figure 4.9: Notifications Module

The other existing project is the Notifications, Figure 4.9, which contains everything regarding the notifications.

It contains the Entities, the Home Pages module which has the Administration Notification Page - similar to the other administrative home pages - and the Notifications Page, which can be accessed by the users in the Notifications Role, present in this project's Roles module.

There is also the Libraries and Schemas module used by the Web Services one (similar to the previous projects), containing only BPM and Internal web services, because this is a smaller project that uses fewer components and most of them are retrieved from the Common Parts (such as external message web services).

It is also presented a Calendar module, which contains the calendar used to send the notifications (for example, emails can be sent every day from 8 am to 9 pm) and also the calendar exceptions, that limit it, such as holidays.

This project also depends on the Procedures Project, because it retrieves the procedure's information, such as citizen information, to create the notification.

4.2.5 (External) Citizen Application

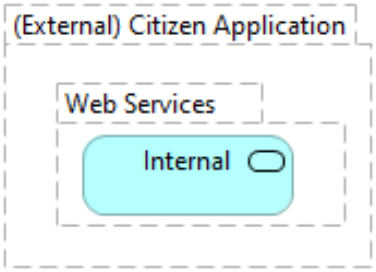

Figure 4.10: (External) Citizen Application Module

The last project is the (External) Citizen Application, Figure 4.10, which is presented in the reference architecture but not created in the application, since it represents, as mentioned, the Citizen Application.

The goal is to provide a better understanding of the architecture, representing it with the components that are useful to the Employee Application, which are the web services.

After this section, all the relevant information to create an application is explained. Important to note, again, that the information here is not final and should be "tailored" to the government/company use cases and the application they have chosen to implement it. Also, there are some modules not represented such as email configurations or decision tables because they are more specific to the government use cases, the application chosen, or because they are not created by the developer but automatically by the application.

4.3 Application Implementation

This section is divided into two parts. The first part consists of a brief introduction regarding the software used in the creation of the application. The software here presented was chosen simply because it was what the company that performed the DT of the government used (and from that, the information was generalized and adapted to this dissertation, as mentioned in the previous chapters).

The second part goes in-depth about the different components that the application contains and how they can be used to implement the models previously proposed. It is relevant to mention that most of the components mentioned here, such as business process models or entities, are also present in other low-code software applications used by other companies (as seen in the Related Work chapter), and because of that, the information presented can be used even if the chosen software differs from the one mentioned here.

4.3.1 Introduction

AppWorks [28] is a software from the company OpenText [27], which focuses in information management software. AppWorks allows the Rapid Application Development (RAD), which is ”an agile software development approach that focuses more on ongoing software projects and user feedback and less following a strict plan” [48]. This user feedback-oriented approach is very important in the DT of organizations because it eases the many concerns that they might have. To achieve that approach it is necessary, however, that the software contains low-code components such as entities and forms.

Also relevant is the integration with other software namely Content Management and Active Directory (which are present in the OpenText solutions, mentioned further). Other similar software, such as OutSystems, also contain these applications and other low-code components similar to the OpenText [12]. Given this, even if the information mentioned here is presented in a generalized way, there are cases and terms which might be specific to the software chosen and, as such, should be adapted if necessary.

4.3.2 Creation of the Application

Before mentioning Appworks and its components, it is important to briefly explain what are the other OpenText solutions, Directory Services, and Content Server and how they relate to the AppWorks platform.

4.3.2.A OpenText Solutions

Starting with the OpenText Directory Services, [49], (also known in other software applications as Active Directory), it consists of the management of ”the users for a single sign-on (SSO) to the OpenText Enterprise Information Management” [50]. This means that each employee user does not need to sign in to every OpenText application. It also helps with the roles module, which will be explained later.

The second important solution is the OpenText Content Server, [51], (also called Content Management), which is the ”content repository and...document management technology...giving control over documents and business content...by securing and storing it throughout its lifecycle” [52]. This is the solution used to store the documentation related to the citizens, the processes, notifications, and others. It also relates to permissions which will also be explained later.

There are also other solutions present in OpenText that can be used to perform the DT, but since they were not used in this specific case, they will not be mentioned. Companies, however, should use them, if deemed necessary and they possess the knowledge.

4.3.2.B OpenText AppWorks

Having mentioned the other two main solutions, AppWorks is now presented. This sub-section tries to make a parallel between the modules and components in AppWorks and what was presented in the previous sub-sections, to explain how to use AppWorks to implement the application from the information provided.

The original application implementation, which this dissertation takes as a foundation, can be extrapolated to the implementation and adaptation of the models and reference architecture provided.

To create the various projects, inside a Collaborative Workspace (CWS), first, it is necessary to configure and create an organization. Organizations or tenants are containers for ”items of your organization such as users, domains,...” [53]. This is where all the modules, services, and other components related to the organization are stored. In this case, AppWorks has the ”ability to configure multiple organizations” [54].

Inside an organization, it is possible to create a CWS, Figure 4.11, which is where ”all modeling activities are done” [54]. This is where the different projects, entities, BPM will be and where the developers can work simultaneously and in a single repository.

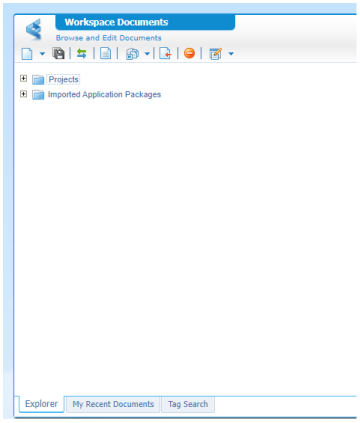

Figure 4.11: Collaborative Workspace

Inside the CWS there is a folder called Project which will contain all the different projects mentioned earlier: Common Parts, Procedures, Notifications, External Entities, and others if necessary.

After creating each project, it is necessary to create the different components inside of it (such as entities or home pages). This is done through a file system, Figure 4.12, where each sub-folder from the project corresponds to each module mentioned.

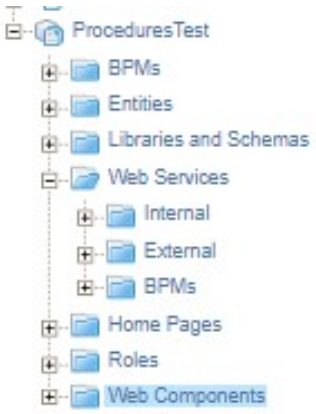

Figure 4.12: File system

Regarding the various projects, the folder Entities contains all the entities mentioned in the reference architecture and Chapter 4.1.1.. Each entity (or business object) contains various building blocks, Figure 4.13.

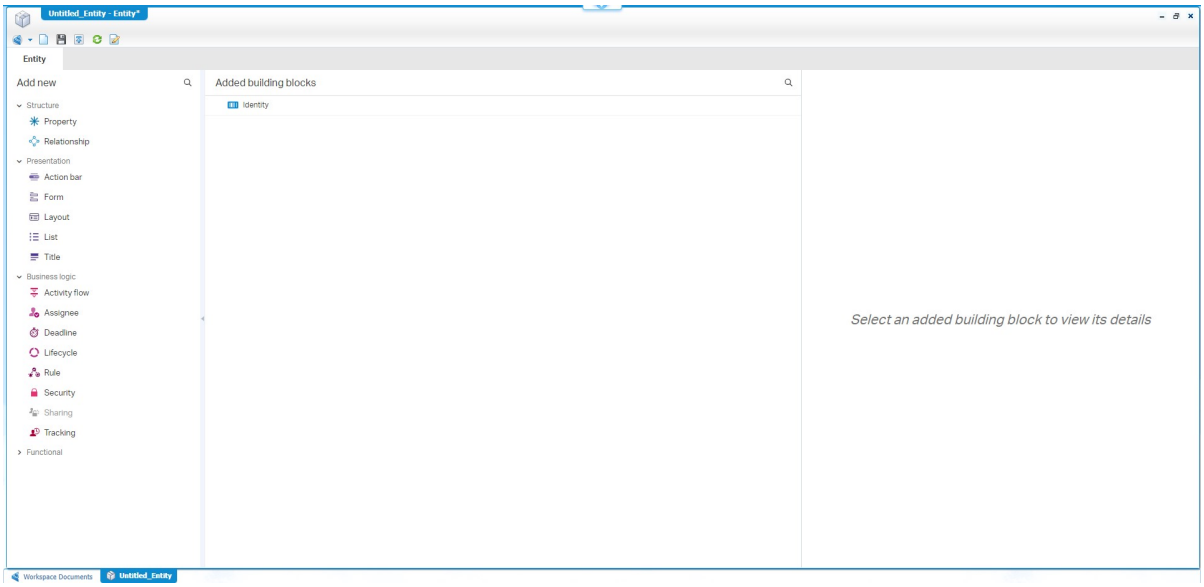

Figure 4.13: Entity Object

The ones relevant to mention are:

- Properties: Properties belonging to the entity such as Payment Amount or Departments.
- Relationships: Entities can have relationships with other entities, for example, parent-child or one-to-many.
- Security: Each entity can have certain permissions to create, view, utilize, and others, which depend on user roles.
- Forms, Layouts, and Lists: These are a few building blocks that are used to show information in the user interface (to the employees).
- Lifecycle: ”Adds case management functionality to an entity” [54]. Case management helps ”manage...unstructured an typically most expensive and complex processes” [55], such as ”user driven, ad-hoc, use cases” [54].
- Web services: Allows the choice to include/utilize a set of web services related to the entity (create, read, update and delete) or to create new ones with the usage of an editor.
- Business Workspace: Enables the connection of the entity to the Content Server business workspace, allowing the synchronization of properties [54].
- Email: ”Adds the ability to send and receive emails associated with an entity instance”. It allows communication such as with citizens, or external entities.

The transformation from the Domain Model to the Entities folder in AppWorks is direct, only requiring the creation of the Entity (right-click in the folder, new, entity), and addition of the necessary building blocks mentioned earlier.

Specifically, the building block Lifecycle, Figure 4.14, can use the sub-section 4.1.2. information regarding lifecycles to ease its creation, given that both share the layout, only requiring the addition of BPM, activities, and specifying the transitions.

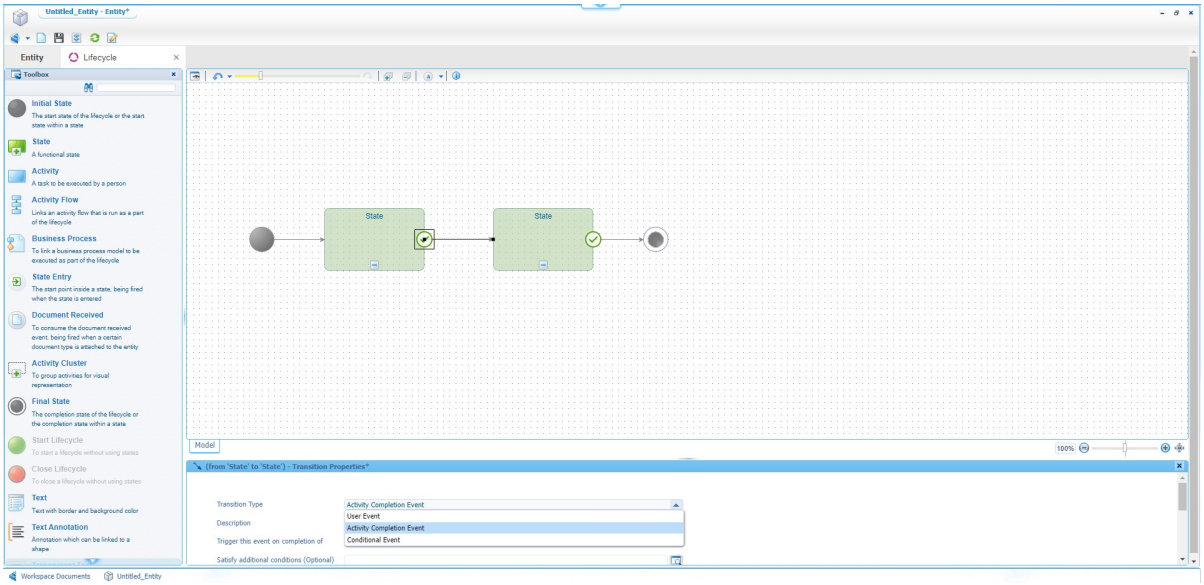

Figure 4.14: Lifecycle Editor

Regarding the BPM module, there is not much information in the previous chapters because Business Process Management is used to model ”well-defined, structured, processes” [54]. Meaning that the business process models are private and specific to the Government in question, requiring them to be created accordingly to the government’s goals.

Despite that, it is important to mention, briefly, the various components in a BPM and how they operate. BPM editor, Figure 4.15, contains activities, which can web services, user layouts and other BPM.

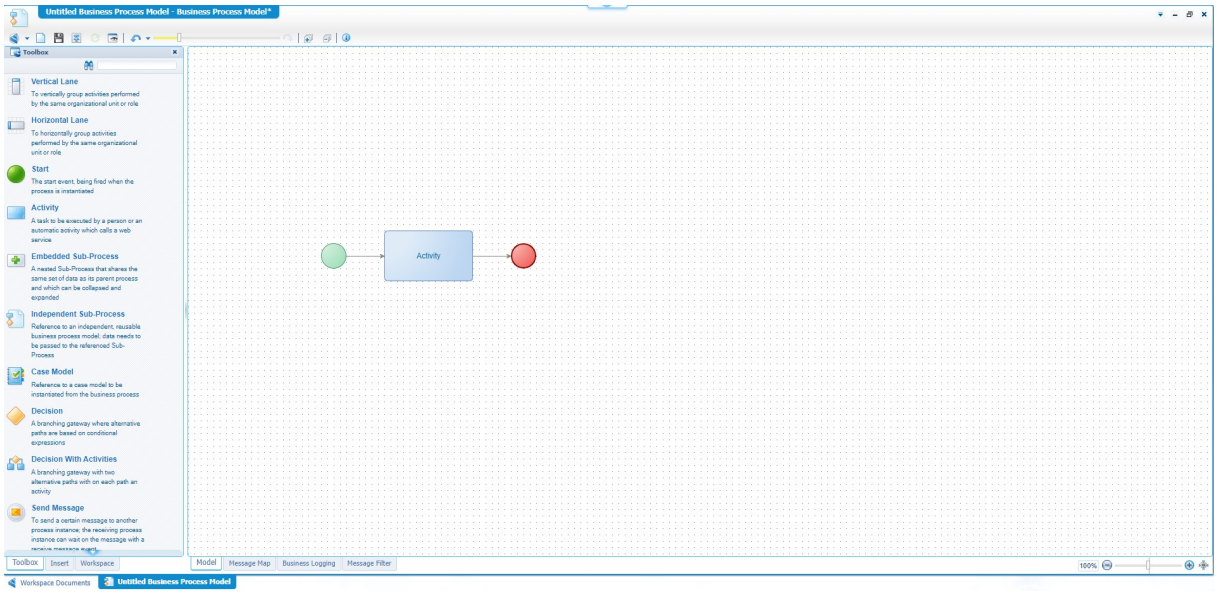

Figure 4.15: Business Process Model Editor

These activities can contain permissions such as only allowing specific roles to perform them.

There are also decisions, which can be exclusive, inclusive, and parallel, based on an input condition, which decides what path to take.

Lastly, there are the cycle blocks, such as Until or For Each. These blocks allow the execution of what is inside them until the termination condition is met.

There are also other components such as Decision With Activities, but these are the most commonly used and vital to the creation of the BPM. Types of BPM can be ”Creation of a Procedure”, ”Creation of a Notification” or ”Creation of the Resolution Document”.

Also important, as mentioned before, is that a BPM can be transformed into a web service and used in other projects and applications.

Regarding the Web Services module, there are four types possible, as presented before:

- Internal: Web services that are created through the entity building block Web services or standard from AppWorks such as ”Retrieve All Users”.
- External: Web services that are created in AppWorks from external connections to other applications (namely web services from the citizens’ application, or OTCS web services).
- BPM: Web services created from existing BPM. To change these web services is necessary to edit the original BPM and generate again the web service, replacing the existing one.
- Java: Web services that come from external files (jar in this case). This means that Web services that are not available in AppWorks can be created in Java and then migrated to the application, through the usage jar files.

Web Services depend on the module Libraries and Schemas that are used to realize them and can also depend on the module Java Metadata (which contains Java definitions) if that is the case. Web services can also be easily tested given that AppWorks provides an application, Web Services Explorer, which allows an easy way to search, see their information, and test them.

The other module is Roles. Each user can be assigned one or more roles in the application through a specific component called ”User Manager”, Figure 4.16.

This component contains the roles standard in the AppWorks (administrator, user...) but can also contain roles created inside the projects. These roles have certain permissions and can be used in the entity’s building block security to restrict the actions that correspond to that entity.

Other possible components that roles can restrict are BPM, web services, and home pages. These roles can be specific to a project or global (for example, if created in the Common Parts and used by the other projects). There are also roles only related to an instance (such as a procedure), which are useful when sharing/delegating instances of entities.

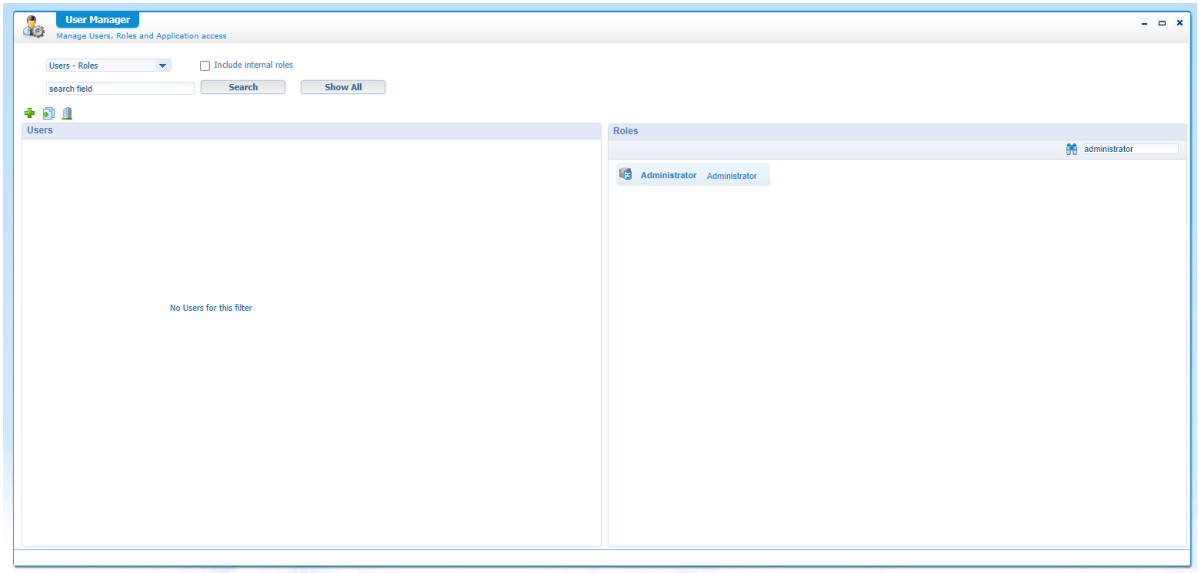

Figure 4.16: User Manager Component

The last modules are the Home Pages and Web Components. AppWorks is divided into two parts: the AppWorks Platform, which is where the CWS, projects, applications such as User Manager or Web Services Explorer, and others are present. This is where the developers and application managers can create, manage and control the application (the back end).

There is also another part called AppWorks Experience (the front-end), Figure 4.17, which is the user interface where the employees perform their tasks.

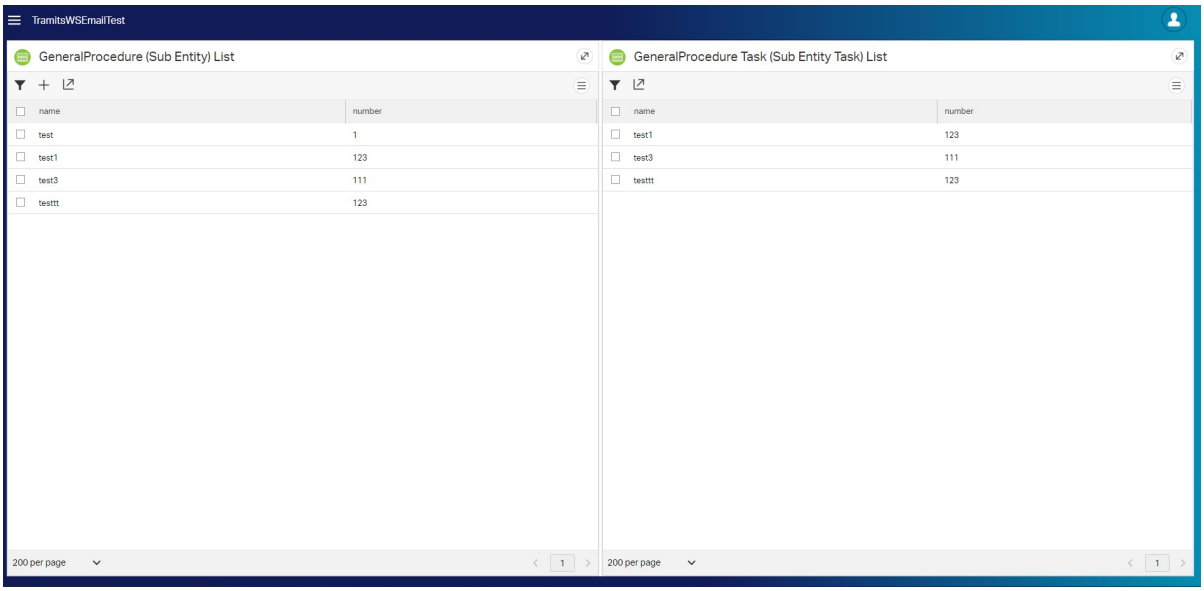

Figure 4.17: AppWorks Experience

For the employees to see and interact with anything, the module Home Pages must be created, containing the various web pages such as Procedure or Administrative pages. These are created, simply, by using the various blocks, for example, the layouts or lists from the entities created, in the home pages components, Figure 4.18.

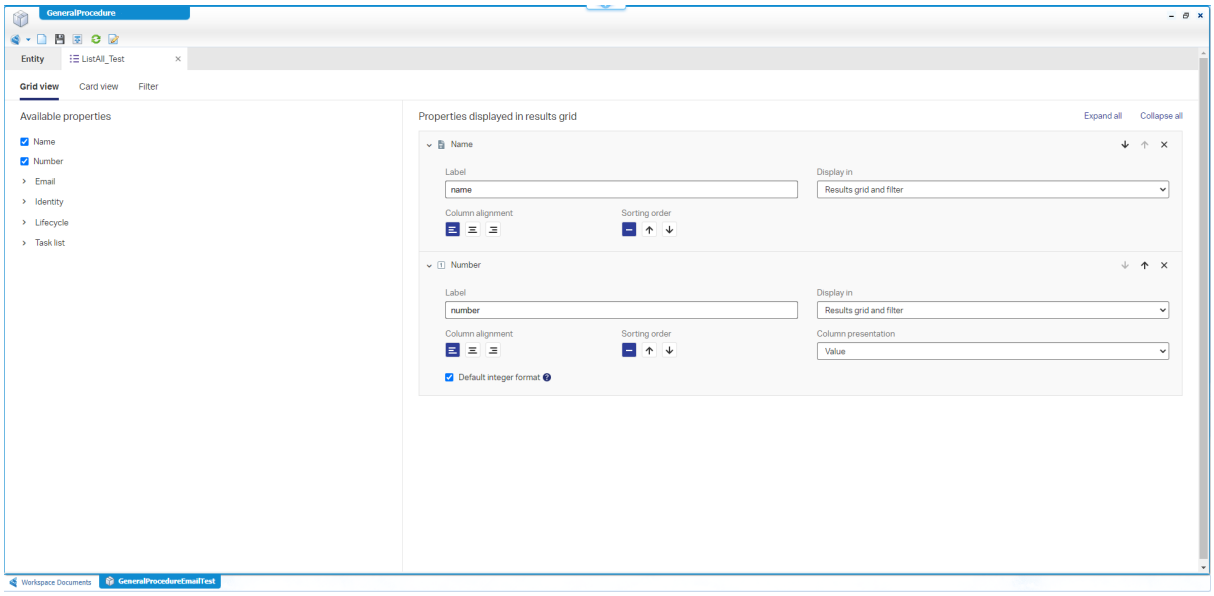

Figure 4.18: Entity List Component

Home pages can also use external components such as Web Content (links to URL pages or Google Maps API, for example), while also allowing the usage of external pictures stored in the Web Compo-

nents module, such as logotypes or watermarks.

All the information presented in this chapter aims to demonstrate how to and how was created the application, from the models and reference architecture provided.

Not only that but the idea is also to show that AppWorks allows the creation of applications faster and easier than using ”traditional software”, presenting an alternative to those methods.

Having created the application, the following chapters provide the evaluation and conclusion of what was presented until here.

5

Evaluation

Contents

Since the development of the real-world application was performed in a government and the models/reference architecture mentioned are niche to this type of transformation, their evaluation cannot be quantitative, such as multiple user tests or online questionnaires.

Explaining better this limitation, user tests are not allowed outside the scope of the government, and since the application just reached the production phase, the employees are still yet to test it. Also, even if they had tested it already, obtaining authorization to question them might not have been possible. Adding to this, since the company's development group that performed the DT is very small, with only 5 developers (one of the benefits of RAD and the usage of EIM/low-code technologies), few developers are knowledgeable about these topics. Because of this, there are very few participants that can be used to evaluate the solution proposed, potentially making the "results ... not be statistically reliable" [56].

Instead, the method needs to be qualitative, where the focus will be not on the number of answers but their quality. [56] mentions that "the best results come from testing no more than 5 users" if the testing is done qualitative and not quantitative [57] and [58], also adding, from the research done by Jakob Nielsen and Tom Landauer [59], that "by doing a qualitative test with 5 participants, you will identify 85% of the issues". So, even if the evaluation is only qualitative and with only 5 participants, it is enough to make conclusions.

This chapter is divided into two parts. The first part presents the methods chosen to evaluate the proposed solution and the results and discussion from that evaluation. The second part presents some previous papers that handle similar transformations and relates them to this dissertation, such as, if what is here presented could help them in their transformation.

5.1 Evaluation Methods

This section is divided into two sub-sections:

- The first one evaluates the models and reference architecture by interviewing some experts and juniors in the area about what they think of them, such as how well they can translate them to a real-life scenario, easiness of understanding and improvement, flexibility and scalability, and others.
- The second sub-section resorts to the developers that created the real-world application mentioned and questions them regarding the usage of low-code technologies and EIM solutions.

5.1.1 Experts/Juniors Evaluation

This evaluation involved an interview with five participants, where they each answered five open questions, to better understand the advantages and limitations of the models and reference architecture here

presented. The distribution of the participants is three experts in the field of digital transformation of processes and RAD, and two juniors that work in the same area.

The questions were the following:

1. Is the information presented here enough to give a ”head start” or to aid in the DT of governments?
2. How easy it is to interpret what is mentioned here, from a junior’s perspective?
3. Are the models and reference architecture flexible and scalable enough to be adapted in real-world scenarios?
4. What are the main limitations and flaws in the models and reference architecture presented?
5. Would you use what is here presented in your next DT of processes?

Regarding the questions, 2. is exclusive to Juniors and 3. to Experts (given the complexity of the questions). The rest of the questions can be answered by both groups, even if the questions are more directed towards a group (for example questions 4. and 5. being more towards experts, than to juniors).

The idea to have two groups, instead of only experts, is to identify if the information presented here is easy to understand, while also providing another viewpoint from people with lower knowledge of the area. This can be extrapolated to government personnel, that also do not understand the area as much as an expert.

5.1.2 Developers Evaluation

This subsection focuses on the development of the application. The goal is to identify the advantages and disadvantages regarding the usage of low-code applications (in this case OpenText AppWorks). Here, it was also created a questionnaire to be used in an interview, done with four developers (that performed the DT), with two being experts/seniors and the other two, juniors. The questions were:

1. How long did it take and how easy was it to create the application from start to finish?
2. What were the limitations or problems faced when using that software, and how easy was it to solve them?
3. Regarding the feedback-oriented approach, what were the main advantages provided by using the software low-code?
4. If given a choice, would you choose to use the low-code software or resort to a more ”traditional” way?
5. After reading the models and reference architecture here presented, do you think it would have aided in the creation of the application? How?

Similar to the questionnaire in the previous sub-section, some of the questions are more targeted to the expert/senior developers than to the juniors but despite that, having different groups can provide different views and experiences when developing the application.

5.2 Results and Discussion

This section will also be divided similarly to the previous one but now only presents the results obtained, commenting briefly on them. Regarding the first two evaluation methods, interviews were conducted online, via Zoom calls, where the participants were alone and did not have prior knowledge of what was going to be asked. Also noteworthy that although ”open”, each interview followed a questionnaire, as mentioned earlier. Regarding each question in the questionnaire, the participants were also asked to evaluate what was asked on a scale from 1 to 5, with 1 being the lowest possible score and 5 the highest. There are a couple of questions that could not be translated to a five-point scale and because of that, only the qualitative part was done.

5.2.1 Experts/Juniors Evaluation

Regarding the first question, about how helpful the information provided was, both juniors report a 4, with two experts reporting a 4 and one 3. Juniors mentioned that despite not knowing enough, the information presented is similar to real-world scenarios that they know, regarding the DT of government processes. The experts explained that although the information presented is not enough, it is useful to help juniors create project demos for customers/governments and to better explain to the government personnel what are the goals of the project.

In the second question the juniors, in how easy was to interpret the information provided, one junior reports a 4 and the other a 5. Juniors mention that the information is easy to understand and to follow even if they lack experience in these types of transformation, which can be extrapolated to government personnel that does not have the technical knowledge but a generalized one.

In the third question, how flexible and scalable the models and reference architecture were, the three experts reported a 4. They realize that the models and reference architecture allow scalability and flexibility enough to mold their specific use cases but realize that some DT implementations cannot fully use them (for example if the implementation is not project/module based, usually ”traditional ways” using only AppWorks for the process management part). When using a low-code approach, since the modules are separated and individualized, it is easy to create or adapt the information according to the necessities (such as names, properties, and others). Also, the lifecycles and use cases can be molded and improved easily enough.

In question 4, juniors mention that the BPM modules lack information, for example in the lifecycle and reference architecture, and having generalized ones could help improve and better understand the information presented. Regarding the experts, they mention that the information presented is not new to them (experts) but it is good to have it modeled since they lack knowledge of the existence of such models. They also mention that not every DT of processes involves the full use of the low-code technology, having for example parts where traditional code is preferred or using frameworks that substitute the AppWorks home pages. Sometimes AppWorks is only used to process the business process models and a few more things. This means that information regarding lifecycles or entities would be irrelevant.

In the final question, if they would use what is here presented, two experts reported a 3 and one a 2, with both juniors reporting a 5. Experts said that they would find a use for the information provided, but mainly by telling juniors to use that information when performing autonomous work, such as creating demo projects to show to the clients (which relieves experts to do more complex tasks). Experts also mentioned that despite the information being ”common sense” to them, they would use it to make the DT more unified and formatted and that is nice to remind them of future problems that can occur (for example circular references).

The Figure 5.1 summarizes the findings.

Questions	Juniors	Seniors
1. Is the information presented here enough to give a “head start” or to aid in the DT of governments?	• Mention that despite lack of knowledge, the information; presented is like the real-world scenarios regarding the DT of government processes.	• Although the information presented here is not enough to create a full fleshed project, it is useful as a base, and it helps juniors create project demos for governments; • It is also useful to better explain governments what the DT consists.
2. How easy it is to interpret what is mentioned here, from a junior’s perspective?	• Despite not having much experience in this type of transformation, the information is easy to understand and to follow (this can be extrapolated to government personal).	• Not asked.
3. Are the models and reference architecture flexible and scalable enough to be adapted in real-world scenarios?	• Not asked.	• The information presented allows scalability and flexibility enough to mold their specific use-cases and adapt the RA and models provided; • Few DTs cannot fully use the models and reference architecture provided.
4. What are the main limitation and flaws in the models and reference architecture presented?	• Mention the lack of specification, mostly and detail, for example in the BPMs or in the lifecycles regarding the transitions.	• The information provided is not “new”, but it is good to see it modeled and researched since they are not aware of similar studies; • Mention that not every DT can take fully advantage of low-code technologies.
5. Would you use what is here presented in your next DT of processes?	• They would use it, because of all the benefits already mentioned.	• They would recommend it to Juniors, for example when performing demo projects to show to the clients; • They would use it to make the DT more standardized (with the adding bonus of preventing problems that sometimes occur).

Figure 5.1: Summary of the Findings

The Figure 5.2 presents the scores relative to the questions asked. The mean obtained was 4.04 with a Coefficient of Variation = 6.79454%, approximately 4 out of 5, meaning that the models and reference architecture presented to the participants are indeed relevant and useful to them.

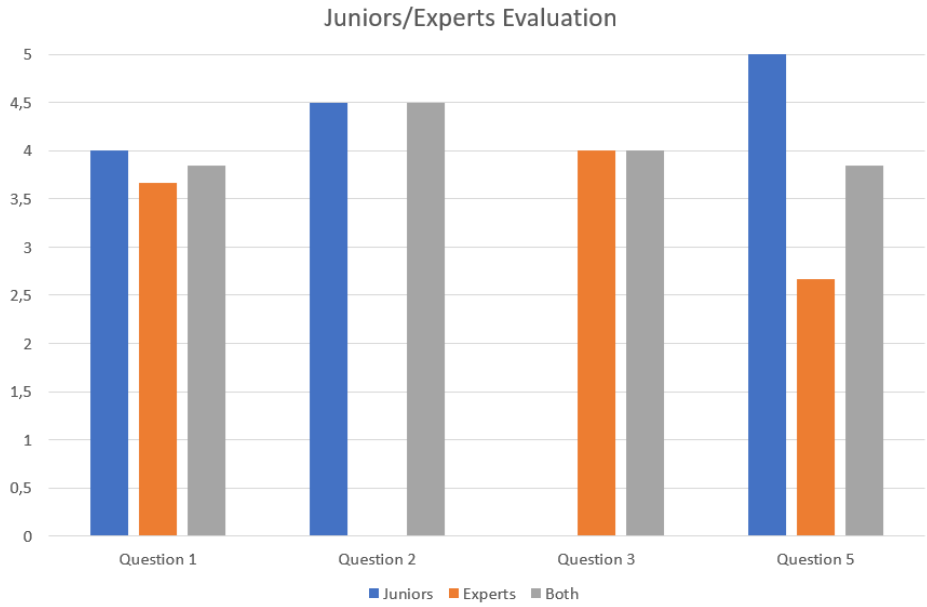

Figure 5.2: Scores from the answers to the questions proposed

5.2.2 Developers Evaluation

Regarding the developers of the application, in the first question, they mention that the development (from developing to production, not counting the after-fixes or maintenance) was roughly one year, with the project changing a lot over time from the initial planning. Experts consider the time of development fast, relative to resorting to traditional ways only. Quantifying the easiness to create the application, one junior reported a 4 and the other 5, with both experts reporting a 4. The majority of the problems encountered were regarding the system's limitations (such as bugs or lack of features) and the number of adaptations over time, from the initial request. It is also important to notice that very few human resources were needed, requiring only one expert and two juniors (there was also another expert but it was more on the bureaucrat parts and not so much in the development).

In the second question, both groups report that the majority of problems were related to the application and its limitations (such as bugs or other issues). Regarding how easy was to solve them, one junior reported a 3 and the other a 4, and both experts reported a 3. As mentioned, the issues faced, were most of the time regarding bugs with the software used, which would require the experts to communicate with OpenText to fix them. Also, the lack of knowledge about the use of the software in a project of this magnitude (even if experts already had some from other projects) and not accounting for possible

future problems such as not taking into consideration the need for more projects (which happened with notifications, that were not in the initial planning), causing circular errors very difficult to resolve. Also, a common problem was that the debugging was hard and the error messages generic, adding to the difficulty and being time-consuming).

Regarding question 3, since the development was faster and could be done in real-time with an interface, the clients were satisfied, felt more inclusive, and could provide faster and optimized feedback, which lead to, as previously mentioned, various changes in the application's goals and features from the initial plan. This is better for the clients but worse for the developers that need to constantly change things, which is why an application (or models/reference architecture followed) needs to be flexible and scalable. Regarding the clients, it allowed them to be happy with the product (which could lead to the company being responsible for the project's next phase), and not dependent on the final stages of development to provide feedback and changes which could be complex, cost money, time, and sometimes impossible to implement without having to create a new project.

In question 4, if they would use Low-Code again, all the participants, unanimously, chose 4. Both groups agreed that using the low-code software to develop this type of application was very useful and they would choose to use it again if given the choice (even with the problems previously mentioned). It was faster and easier, and despite the learning curve, allowed the juniors to work semi-autonomous and without major issues. They note, however, that "traditional ways" allow a better understanding of the processes behind the development and it is easier to identify the errors that appear (adding to the difference in the quantity and quality of the information available on the web, books,.. regarding the "traditional ways" compared to the usage of AppWorks). Mentioning this, they understand that a balance regarding the usage of low-code and traditional ways is vital to improving the development of applications.

Regarding the final question, if the information presented would have aided them, the two juniors chose 4 and 5, and both experts reported 4. The two groups agreed that having used the information here presented would have accelerated the development of the project (because some projects were requested in later stages of the application and there was plenty of free time in the initial stages that could have been used to create them). Also, problems that came from not having a common project containing the components used in multiple projects created several issues such as circular references. Another problem is that they had to create the same components in several projects (which could have been prevented). Having this information would have allowed the creation and presentation of a demo in the early stages, such as planning, and provided even more initial feedback from the government.

The Figure 5.3 summarizes the findings.

Questions	Juniors	Seniors
1. How long did it take and how easy was it to create the application from start to finish?	• Development took roughly a year (this first phase, which the company was responsible for)	• Development time is deemed fast (compared to using "traditional ways"); • They note that the project changed a lot over the time, which was made possible and facilitated because of the usage of low-code technology; • Refer also that the very few human resources were necessary, only requiring a group of 4/5 people.
2. What were the limitations or problems faced when using that software, and how easy was it to solve them?	• Report that the major problems faced were regarding the application used, such as bugs or limitations); • Lack of knowledge regarding the usage of the software, in a project of this magnitude	• Adding to what was said, not accounting for possible future problems, which originated circular dependencies, difficult to resolve; • Debugging, sometimes is hard because the details of the error are not specific.
3. Regarding the feedback-oriented approach, what were the main advantages provided by using the software low-code?	• The clients were satisfied because they felt inclusive in the development of the application, since they could provide a lot of feedback which was treated rapidly.	• The feedback oriented approach, although more demanding on the developers was better then just receiving it in the end, when making major changes could be complex, time consuming and so on.
4. If given a choice, would you choose to use the low-code software or resort to a more "traditional" way?	• They agree that using low-code software has a lot of advantages, and would use it again, given the choice; • Mentioned that, however, would still like to use "traditional ways" when necessary, if possible. • AppWorks is easy to learn, compared to the traditional ways, even if they allow a better understanding of the application (and better debug sometimes)	• Same opinion as the juniors, using low-code is preferred even more because the ability to integrate the traditional ways (programing languages) is possible, at least in AppWorks; • Management side, such as controlling users or processes is also easier.
5. After reading the models and reference architecture here presented, do you think it would have aided in the creation of the application? How?	• Both juniors and experts agree that having used the information provided would have accelerated the development of the project, since it would have mitigated the problems faced.	• Apart from what was mentioned, it would have allowed the juniors to faster and easier create a demo in the initial stages, allowing even earlier feedback.

Figure 5.3: Summary of the Findings

Important to refer, again, that this application implementation was the one that served as a foundation for this dissertation and not an adaptation/implementation of the information provided here (models and reference architecture) in a real-world scenario. Despite that, and since the solution proposed is based on in, it is possible to, partially, extrapolate it to an adaptation and implementation of the information provided. This, however, does not substitute a real-world implementation of the information provided, which is mentioned in the Future Work chapter.

The Figure 5.4 presents the scores relative to the questions asked. The mean obtained was 3,94 with a Coefficient of Variation = 10.40863%, approximately 4 out of 5, meaning that, similarly to the juniors' and experts' evaluation, the usage of low-code technology and the feedback-oriented approach is very important and useful to the implementation of this type of DT.

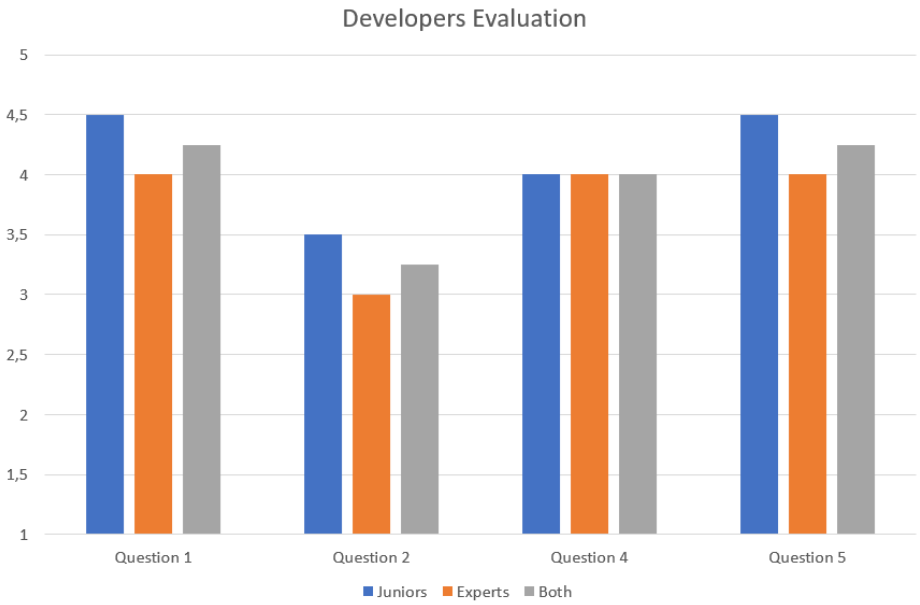

Figure 5.4: Scores from the answers to the questions proposed

5.3 Comparison to other Papers

In this final evaluation sub-section, the goal is to compare this dissertation to other papers that mention similar DT, from the related work, and, as mentioned, understand if they could take advantage of the solution proposed here.

5.3.1 Related Work Assessment

The two papers chosen are Mazumder A. [34] and Liu C. et al. [2]. Only these papers were chosen because they are the most identical to this dissertation, containing a DT of processes and a development of a real-world application. The idea is to:

1. Compare the theory and development to the other papers and extract information from them to enrich this dissertation or improve it in the Future Work chapter.
2. Understand if those papers could have used this dissertation to help them, for example, while developing the application.
3. Understand if their limitations could be fixed or mitigated with the usage of low-code software (if that is the case).

The goal is to understand if the proposed solution is useful to other scholarly papers and real-world scenarios, and if not, identify those flaws and correct them in the future.

It is important to note that since the DT of government processes is a very specific theme and without much scholarly information (either theoretical or practical), the papers used are not specifically related to the DT of government processes but a specific DT of eGovernance services.

[34] mentions the creation of an eGovernance application called MARREG, which helps with the handling of marriage certificates. The paper, however, misses the technical parts of the application (how it was created or specific technical components) and instead focuses on the needs and benefits of adopting said technology, mentioning a very generalized solution architecture that contains the major roles and parts. Despite that, it mentions that the application follows specific practices and standards which could be useful to provide more authenticity to this dissertation, in addition to the identification of the benefits that come from its use.

The second paper contains the implementation of a system based on the three-tier B/S architecture [2] which comprises the Presentation Tier, the Business Tier, and the Data Tier. This system uses 4 major components which are the login module, the approval flow module (the processes modulation), the document management module, and the system management module (users, roles...). It does not go in-depth regarding each one, however, and instead correlates the three tiers with those components. The system, contrary to the one here presented, was also created without the usage of low-code applications or case-management ones which caused several issues that will be mentioned later.

From the second question, all of the papers would take advantage of using what is mentioned here. First, it helps them create the application and put the theory into practice. Regarding the paper [2], which created a system, it would have allowed an easier and faster development, with also detailing the flow module (with the reference architecture) and how to relate all the modules proposed by them.

Second, it would have allowed more detail in the information provided. It would have allowed [34] to better explain the processes and rules performed by the application, through the usage of models and reference architecture presented here, not having to rely on the written text that can be confusing.

The last question is also positive for both papers. [34] does not refer to what technology was used to create the system or what models they followed. Since what is presented here is relevant to the DT of government processes, and marriage certificates can be assumed as a government process, the solution proposed in this dissertation could have provided not only a faster and simpler way to create the system but also an improved feedback approach in its development.

Regarding the paper [2], it mentions several issues like only having implemented the basic functions because of ”time, energy, and ability”. This could have been prevented if the authors had followed the models and reference architecture presented here and used a low-code approach when creating the application. Several concerns such as data encryption or network security, or even further functions like decision-making support and mobile office automation, could all be dealt with using what is mentioned in this dissertation.

The Figure 5.5 summarizes the findings.

Questions	Paper 1: Mazumder A.	Paper 2: Liu C. et al.
1. To compare the theory and development to the other papers and extract information from them in order to enrich this thesis or improve it in the future work.	• Mentions practices and standards used, as well as the identification of the benefits of using them.	• Mentions the B/S architecture and the 3 tiers which could better group what was presented in this dissertation.
2. Understand if those papers could have used this paper to help them, for example, while developing the application.	• Helps them create the application and implement the theory.	• Helps them create the application and implement the theory; • Would have allowed a better explanation of the information presented with the AR and models.
3. Understand if their limitations could be fixed or mitigated with the usage of low-code software (if that is the case).	• The technology used to create the application was not mentioned but using the one presented here could provide all the benefits already mentioned.	• Many of the issues reported, such as lack of time or complexity could have been prevented by using the solution proposed here.

Figure 5.5: Summary of the Findings

The results show that even if the sample size regarding users or papers compared was very small (because of the constraints mentioned), they confirm that the information/proposed solution presented in this dissertation helps experts and juniors perform the DT, mitigating some errors that commonly occur, with the bonus of helping ”non-tech” people, such as government personal, to understand better the project and be more present in its development.

6

Conclusion

Contents

This last chapter presents the conclusions regarding the whole project and an analysis of the positives and the limitations/issues found. It also contains the future work that should be done to improve what is here proposed.

6.1 Main Contributions

In conclusion, from the results previously obtained, this dissertation provides valuable information regarding the topic of digital transformation of government processes.

First were identified, from the related work and the real-world implementation, the most relevant use cases: Resolve a Procedure, Reclassify a Procedure, Create a Request, and Send a Notification by Hand. Following that, were created the entity models from the relevant entities identified and their lifecycles. That information was then combined in a reference architecture, also presenting how to implement it through a low-code application.

The reference architecture comprises 5 projects (named after the AppWorks nomenclature) which are the Procedures, Notifications, External Entities, Common Parts, and (External) Citizen Application (with this last one referring to the Citizen Application).

Each project contains several modules which can be:

- Entities (mentioned in the Domain Model);
- Home Pages and Web Components, used in the User Interface of the application;
- Roles, which define the permissions of the users/employees;
- BPMs, where the modulation and realization of the business processes are;
- Web Services, which are services that can be used in the BPMs;
- Libraries and Schemas, which the web services can retrieve data from, especially if they are created externally and imported by the application.

These models and how to implement them were done to provide information, aid the DT of processes, and incentivize the governments to implement or improve their DT.

However, since the real-world implementation that served as an inspiration for this dissertation (which can be considered an implementation of the information presented here) was done in a government (which can not be disclosed, nor specific details regarding that implementation) and the topics mentioned are niche in the field of Information Technology (IT), a full statistical quantitative evaluation of the dissertation could not be performed. Nevertheless, it was performed a qualitative one, which comprised a few interviews with experts and juniors in the field, in order to evaluate the work done through the

opposing viewpoints. Juniors are especially useful because it allows their testing to be extrapolated to people which are related to the topics studied but not experts such as government personnel. That evaluation divided into two parts, with the first one regarding the information presented in this dissertation and the second one regarding the real-world implementation that was done (in order to evaluation the low-code usage).

Adding to the interviews, a comparison with other papers was also conducted where the goal was to identify if they could have taken advantage of using the information presented here. From the results, the major takeaways were the following:

1. The models and reference architecture presented are easy to understand and follow from an expert perspective but also from a junior/low-knowledge one.

2. The information here, despite not presenting new knowledge to the experts in the area, allows them to confirm their development, systematize the DT, and mitigate errors. It would have been very useful if it was presented to them before implementing the solution (which is not strange, given that this dissertation is based on that implementation). It is also useful to juniors or people not familiar with this type of DT because of the provided knowledge.

3. The usage of low-code technologies eases the DT of government processes and allows the mitigation of issues that can come from its implementation.

Given these conclusions, it is safe to assume that what is proposed helps solve the main issues that this dissertation tackles, which are the lack of a reference architecture and associated models used in the DT of government processes and the lack of information regarding the usage of low-code technologies when implementing such transformation (such as limitations).

6.2 Limitations and Future Work

The following sub-section is divided into two, Limitations which focus on the problems and limitations identified, and Future Work which, as the name suggests, mentions possible improvements to be done in the future.

6.2.1 System Limitations

Regarding the limitations, several were already mentioned. One of them is that these models, reference architecture, and implementation are optimized to be used by low-code technologies, specifically OpenText AppWorks. This is because of the nomenclature and the different models that, although can be adapted to other technologies, were created with that specific technology in mind. This also skews the

part of the evaluation development because different low-code technologies can lead to different development duration, problems, and others (even if in a generalized way, those technologies should function the same). Investigation regarding the implementation using other technologies should be done.

Another problem is that this implementation is dependent on a specific version of AppWorks. This means that in the future, for example, there will be better ways to implement the information provided, which makes the implementation chapter not as useful even if, again, the information can be adapted to the new version.

The models or RA are also very specific to this type of DT and might not be useful if it diverges vastly from this one. Also, there are other modules such as Schedulers or xForms that are not present in the reference architecture (because they were deemed too specific) but could/should be. More information regarding the BPMs or more technical components such as Service Groups or types of web services (REST, SOAP) should also be provided.

It is important to mention that further testing is necessary to improve the information here presented. Not only more testing in other real-world implementations but also testing the implementation chapter which, although provides only a skeleton (basics) can be useful in early stages when presenting the idea to the government personnel.

Another problem is that since this information is not new to experts (and is affected by being created during/after the creation of the application and not before), they might not find the need to use what is here proposed, even with all the benefits mentioned earlier (such as changing companies in the middle application development).

Despite those problems, it is believed, given what was previously mentioned, that this dissertation can help fix some issues regarding the main topics mentioned and hopefully help, and incentivize governments to digitally transform their processes.

6.2.2 Future Work

Regarding future work, an improved evaluation should be performed. As mentioned before, a better evaluation was not possible in this dissertation because it was implemented in a government that asked for privacy, and because it was specific to a niche field, few participants were identified which in turn made the evaluation qualitative and not quantitative. In the future, a better analytical evaluation should be performed to provide better conclusions, with also the analysis of the implementation of the information presented here, in other governments or as a project demo.

It could be also interesting to evaluate the government employees and get their opinions and feedback regarding the new technology and compare it to the previous systems.

Another improvement should be to compare the traditional technologies to the low-code ones when implementing the models and reference architecture mentioned. Improvements should be done accord-

ingly to better accommodate both types (since this dissertation focuses more on the low-code one).

Lastly, providing an implementation of the solution proposed using another low-code technology could be valuable, such as comparing it to the one presented here, in order to identify the best one (in this case).

Bibliography

[1] Salesforce. What is digital transformation? — a definition by salesforce. Available: https://www.salesforce.com/eu/products/platform/what-is-digital-transformation/. [Accessed June 29, 2022].

[2] Liu, C. and jun tao, X. and Lu, B., "Toward integrated and automated management of government affairs," *International Journal of Hybrid Information Technology*, vol. 9, pp. 267–278, 01 2016.

[3] Bower, L. L., "Digital transformation of public services and administration: A case study of bulgaria," in *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, ser. ICEGOV 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 267–270. [Online]. Available: https://doi.org/10.1145/3428502.3428540

[4] Liang, P. and Liu, Z., "Realization of administrative organs automation system ased on b / s mode," in *2010 International Conference on Machine Learning and Cybernetics*, vol. 5, July 2010, pp. 2270–2275.

[5] Kimberling, E. (2021, May) How amazon is a roadmap for digital transformation success. Available: https://www.thirdstage-consulting.com/amazon-roadmap-for-digital-transformation-success/. [Accessed July 13, 2022].

[6] Helvetas. Government 5.0: The future of public services. Available: helvetas.org/en/eastern-europe/about-us/follow-us/helvetas-mosaic/article/March2020/Government-5.0-Future-of-Public-Services. [Accessed June 29, 2022].

[7] Coursaris, C. K., "Enabling mgovernment: a framework and a case study," *International Journal of Electronic Finance*, vol. 6, pp. 79–101, 04 2012.

[8] Senyo, P. K. and Effah, J. and Osabutey, E., "Digital platformisation as public sector transformation strategy: A case of ghana's paperless port," *Technological Forecasting and Social Change*, vol. 162, 01 2021.

[9] Singh, J. P. and Kattimani, S. M. and Kumar, R. and Ramanathan, S. and Prasad, G. N. V., "Need aspect approval system for procurement of components used for indian satellite program," in *2018*

4th International Conference on Recent Advances in Information Technology (RAIT), March 2018, pp. 1–6.

[10] Krishnan, G. and Ravindran, V., "It service management automation and its impact to it industry," in *2017 International Conference on Computational Intelligence in Data Science(ICCIDS)*, June 2017, pp. 1–4.

[11] Deloitte. Managing risk in digital transformation. Available: https://www2.deloitte.com/content/dam/Deloitte/za/Documents/risk/za_managing_risk_in_digital_transformation_112018.pdf. [Accessed June 30, 2022].

[12] OutSystems. The low-code development guide. Available: https://www.outsystems.com/guide/low-code/. [Accessed June 30, 2022].

[13] Nugroho, L. E. and Egaravanda, S. and Achmad, K. A., "An architecture for facilitating two-way g2c relationships in public service delivery," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 4, pp. 1179–1184, 2018. [Online]. Available: http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=6515

[14] Eckermann, O. and Weidlich, M., "Flexible artifact-driven automation of product design processes," *Enterprise, Business-Process and Information Systems Modeling*, p. 103–117, 2011.

[15] Velamala, R. R., "Collaborative government to employee (g2e): Issues and challenges to e-government," *Journal of E-Governance*, vol. 34, pp. 214–229, 12 2011.

[16] Lee, J. and Kim, B. and Park, S. and Park, S. and Oh, K., "Proposing a value-based digital government model: Toward broadening sustainability and public participation," *Sustainability*, vol. 10, no. 9, p. 3078, 2018.

[17] Fath-Allah, A. and Cheikhi, L. and Al-Qutaish, R. and Idri, A., "E-government maturity models: A comparative study," *International Journal of Software Engineering and Applications*, vol. 5, pp. 71–91, 03 2014.

[18] Markusheuski, D. and Rabava, N. and Kukharchyk, V., "Blockchain technology for e-governance," ser. The Conference of Innovation Governance in the Public Sector, 2017.

[19] Papadopoulou, P. and Kolomvatsos, K. and Hadjiefthymiades, S., "Enhancing e-government with internet of things," *Computational Intelligence in the Internet of Things*, p. 110–129, 2019.

[20] OutSystems. What is business process management (bpm)? Available: https://www.outsystems.com/glossary/what-is-business-process-management-bpm/. [Accessed September 02, 2022].

[21] Reijers, H.A., “Business process management: The evolution of a discipline,” *Computers in Industry*, vol. 126, p. 103404, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521000117

[22] kissflow. (2021, 11) What are document-centric, integration-centric, and human-centric bpm? Available: https://kissflow.com/workflow/bpm/document-integration-and-human-centric-bpm/. [Accessed August 19, 2022].

[23] jdsupra. (2020, 11) Workflow automation vs process automation: What’s the difference? Available: https://www.jdsupra.com/legalnews/workflow-automation-vs-process-36714/. [Accessed August 19, 2022].

[24] Avliotis, A., “Business process automation and robotic process automation: Early deployment of a combined approach,” 2019.

[25] Engin, Z. and Treleaven, P., “Algorithmic Government: Automating Public Services and Supporting Civil Servants in using Data Science Technologies,” *The Computer Journal*, vol. 62, no. 3, pp. 448–460, 03 2019. [Online]. Available: https://doi.org/10.1093/comjnl/bxy082

[26] mendix. What is low-code? Available: https://www.mendix.com/low-code-guide/. [Accessed August 19, 2022].

[27] OpenText. Opentext. Available: https://www.opentext.com/. [Accessed July 15, 2022].

[28] OpenText. Opentext appworks platform. Available: https://www.opentext.com/products-and-solutions/products/digital-process-automation/appworks-platform. [Accessed July 15, 2022].

[29] PRISMA. Available: https://www.prisma-statement.org//. [Accessed August 20, 2022].

[30] Maragno, G. and Gastaldi, L. and Corso, M., “E-government for public values creation: A systematic literature review,” in *DG.O2021: The 22nd Annual International Conference on Digital Government Research*, ser. DG.O’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 386–397. [Online]. Available: https://doi.org/10.1145/3463677.3463692

[31] HCL SOFTWARE. Available: https://www.hcltechsw.com/domino. [Accessed August 26, 2022].

[32] Aysolmaz, B. and Demirörs, O., “Deriving user requirements from business process models for automation: A case study,” in *2014 IEEE 1st International Workshop on the Interrelations between Requirements Engineering and Business Process Management (REBPM)*, Aug 2014, pp. 19–28.

[33] AMSTAR. Available: https://amstar.ca/Amstar_Checklist.php. [Accessed August 27, 2022].

[34] Mazumder, A., “Marreg – marriage registration [for a better cause],” in *2020 IEEE Integrated STEM Education Conference (ISEC)*, Aug 2020, pp. 1–8.

[35] Karagiannis, D., “Business process management: A holistic management approach,” vol. 137, 01 2013, pp. 1–12.

[36] Pal, S. and Gupta, G. and Biswas, I., “Codezin: braving the startup storm,” *Emerald Emerging Markets Case Studies*, vol. 7, pp. 1–28, 2017.

[37] Paul, A. and Paul, V., “The e-government interoperability through enterprise architecture in indian perspective,” 10 2012, pp. 645–650.

[38] Fajar, A. and Shofi, I., “Service oriented design for indonesian e-government system using soa,” *IOP Conference Series: Materials Science and Engineering*, vol. 598, p. 012106, 09 2019.

[39] Mahmoud, Q. H. (2015, April) Service-oriented architecture (soa) and web services: The road to enterprise application integration (eai). Available: https://www.oracle.com/technical-resources/articles/javase/soa.htm. [Accessed August 27, 2022].

[40] Gartner. Gartner magic quadrant & critical capabilities. Available: https://www.gartner.com/en/research/magic-quadrant. [Accessed August 29, 2022].

[41] Rosenquist, C. J., “Entity life cycle models and their applicability to information systems development life cycles,” *Comput. J.*, vol. 25, pp. 307–315, 1982.

[42] Russo, M. and Ferrari, A. Avoiding circular dependency errors in dax. Available: https://www.sqlbi.com/articles/avoiding-circular-dependency-errors-in-dax/. [Accessed July 21, 2022].

[43] Learning BPM. Learning business process modeling & improvement. Available: https://bpmn.gitbook.io/learning-bpmn/. [Accessed September 02, 2022].

[44] Ryan. (2020, September) What are libraries in programming? Available: https://www.idtech.com/blog/what-are-libraries-in-coding. [Accessed August 25, 2022].

[45] Dearmer, A. (2021, July) Complete guide to database schema design. Available: https://www.integrate.io/blog/complete-guide-to-database-schema-design-guide/. [Accessed September 10, 2022].

[46] Dhingra, S. (2016, November) Rest vs. soap: Choosing the best web service. Available: https://www.techtarget.com/searchapparchitecture/tip/REST-vs-SOAP-Choosing-the-best-web-service. [Accessed September 13, 2022].

[47] JSON. Introducing json. Available: https://www.json.org/json-en.html. [Accessed September 18, 2022].

[48] OutSystems. What is rapid application development. Available: https://www.outsystems.com/glossary/what-is-rapid-application-development/#:~:text=Rapid%20application%20development%20is%20an,rapid%20prototyping%20over%20costly%20planning. [Accessed July 15, 2022].

[49] OpenText. Opentext: Directory services (otds). Available: https://developer.opentext.com/resources/documentation/3a7bfdae-4cd3-465c-8485-c887f65def9b/a34c843a11f4ff9ea3229dcd116c0f82/page/1. [Accessed July 19, 2022].

[50] OpenText. Enterprise information management (eim). Available: https://www.opentext.com/products-and-solutions/enterprise-information-management. [Accessed July 19, 2022].

[51] OpenText. Opentext content services platform. Available: https://www.opentext.com/products/content-services-platforms. [Accessed July 15, 2022].

[52] OpenText. Opentext document management. Available: https://www.opentext.com/products-and-solutions/products/enterprise-content-management/content-management/opentext-document-management. [Accessed July 15, 2022].

[53] Saxton, A. What is a tenant? Available: https://powerbi.microsoft.com/pt-pt/blog/what-is-a-tenant/. [Accessed July 16, 2022].

[54] OpenText. Opentext appworksplatform architecture. Available: https://developer.opentext.com/dxls/api/v1/asset?asset=/sites/ot2dx/assets/white_papers/downloads/AppWorks_platform_architecture.pdf. [Accessed July 21, 2022].

[55] OpenText. Opentext case management. Available: https://www.opentext.com/products-and-solutions/products/digital-process-automation/alternative-process-solutions/opentext-case360. [Accessed July 21, 2022].

[56] Budiu, R. and Moran, K. (2021, July) How many participants for quantitative usability studies: A summary of sample-size recommendations. Available: https://www.nngroup.com/articles/summary-quant-sample-sizes/. [Accessed September 12, 2022].

[57] Budiu, R. (2021, July) Why 5 participants are okay in a qualitative study, but not in a quantitative one. Available: https://www.nngroup.com/articles/5-test-users-qual-quant/. [Accessed September 12, 2022].

[58] Nielsen, J. (2000, March) Why you only need to test with 5 users. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/. [Accessed September 12, 2022].

[59] Nielsen, J. and Landauer, T. K., “A mathematical model of the finding of usability problems,” *CHI ’93: Proceedings of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems*, vol. 25, p. 206–213, May 1993.

Annexes

Title	Authors, Year	Document/ Source Type	Problem	Government Related?	Contribution	Findings
E-Government For Public Values Creation: A Systematic Literature Review	Maragano G., Gastaldi L., Corso M., 2021.	Conference Paper, Conference Proceedings	Despite very successful cases of DT, few organizations realize its true potential	Yes, but not specified	Conceptual framework, based on SLR, for understanding and influencing e-Government implementation and its benefits.	Before the implementation of DT, a set of variables need to be addressed.
Digital transformation of public services and administration: A case study of Bulgaria	Bower L.L., 2020	Conference Paper, Conference Proceedings	Not directly, focus on studying the DT of Bulgaria's Government	Yes, focused in Bulgaria	Research, using Bulgaria's example, on the improvement and accomplishments of using strategic frameworks and policies,... in DT.	Bulgaria is on its way to become a paper-free administration, with higher level of security and reliability systems.
Towards integrated and automated management of government affairs	Liu C., Xiong J., Lu B., 2016	Article, Journal	Despite more and more systems being designed and implemented in DT, few can take into account the changes in government affairs.	Yes, focused in China	Development of an integrated an automated management system for government affairs	None. Lack of time, energy and ability prevented them to finish the program even more doing analysis and findings.
Digital platformisation as public sector transformation strategy: A case of Ghana's paperless port	Senyo P.K., Effah J., Osabutey E.L.C., 2021	Article, Journal	Most of the research is focused on developed economies with stable institutional environments.	Yes, focused in Ghana	Development of a transformational affordance framework and how DT can enable public Transformation.	Interactions between digital platforms and organization's roles generate relational affordances to enable DT.
An architecture for facilitating two-way G2C relationships in public service delivery	Nugroho L.E., Egaravanda S., Achmad K.A., 2018	Article, Journal	Some specific government measures, like E-government applications, still aren't enough.	Yes, but not specified	E-government application architecture that accommodates specific and temporary needs of standard services.	The unified perspective on the two-way relationship allows the integration of software components, having a wide applicability.
Digital Transformation? A Primer for Practitioners	Doukidis G., Spinellis D., Ebert C., 2020	Article, Journal	Although DT is essential, many companies are afraid because of perceived risks, inexperience,...	Slightly, mostly companies in general	The authors show how to get started with DT, by providing case studies, elements, platforms and so on.	None. There is no direct findings because the purpose of the paper, like mentioned, is to give a helping hand in the start of DT. There is no analysis or evaluation afterwards.
Flexible artifact-driven automation of product design processes	Eckermann O., Weidlich M., 2011	Conference Paper, Book Series	Automation business processes in certain domains (product planning...) are not suited for highly creative processes.	No	Propose a methodology for the implementation of supporting workflows.	The novel structure of the model created enables rich flexibility.
IT service management automation and its impact to IT industry	Krishnan G., Ravindran V., 2017	Conference Paper, Conference Proceedings	Although digital transformation is rapidly transforming organizations, managing growth and reducing costs is not easy for IT infrastructure.	Slightly, generally.	Performance of a SLR in the automation scope of ITSM from ITIL process perspective. Also, presents the positives and negatives of the impacts of automation in IT.	Automation can improve the problems presented, but it does come with its challenges. Also, ITSM scope matrix has been developed to validate the automation approach.
Enabling mGovernment: A framework and a case study	Coursaris C.K., Boylan C., Taylor J., 2012	Article, Journal	Companies that want to support mobile devices or duplicate their existing CMS information or repurpose it to a mobile friendly format	Yes, focused in Oakland Country, Michigan	Explores the case related to this approach and proposes a three-stage approach involving PTF, MCP and ALS.	The three components worked together in order to resolve the challenge of increasing quality of service regarding the increase demand for mobile application.
The application of workflow technology in office automation system	Zhang X., 2012	Conference Paper, Book Series	Having an increase in the amount of information causes issues like more processing power required, technologies to deal with that amount,...	Yes, slightly focused in China.	• This paper presents how the application of workflow technology can help in Office Automation System • Discusses how to use WWF technology to develop and achieve automation	There is no findings, the paper simply mentions the COAS and its architecture. However, it mentions that description ability of workflow should be further strengthened.
Need aspect approval system for procurement of components used for INDIAN satellite program	Singh J.P., Kattimani S.M., Kumar R., Romanathan S., Prasad G.N.V., 2018	Conference Paper, Conference Proceedings	Any decision taken in government needs to be recorded and authenticated (which was traditionally done with human ink signature regarding documents. However, digital signature infrastructure is costly and requires additional training.	Yes, focused in India	• Demonstration of implementation of software assisted collaborative decision making towards achieving larger goal of e-government • Proposes the usage of ink signature on prints outs of relevant e-documents for record keeping and authentication until the creation of full-fledged digital signature infrastructure.	The usage of 'e-NAC' has helped in achieving faster turnaround time and quicker decision process.
Deriving user requirements from business process models for automation: A case study	Aysolmaz B., Demirors O., 2014	Conference Paper, Conference Proceedings	Relations between business process models and requirements are either not established or only partially available. This leads to an increase in effort, broken traceability, completeness and consistency problems for user requirements.	Yes, as part of Turkish Republic of Northern Cyprus - Cyprus	Propose a unified business process modeling method, UPROM, to analyze and develop models for business processes and user requirements	Results show that by applying UPROM, user requirements and business processes can be analyzed in a unified way and textual requirements can be generated automatically.
Realization of administrative organs automation system based on B/S mode	Liang P.-X., Liu Z.-Y., 2010	Conference Paper, Book Series	Government organs are complex and require many communications with other organs or entities. Staff is also limited and requires training.	Yes	• Introduction of the overall design and application design of Administrative Organs Automation System. • Discussion the resolution of its central technology	The system is featured with friendly interfaces and flexibility in operation, which can greatly improve work efficiency and realize a "paperless office".
Adopting DevOps in the real world: A theory, a model, and a case study	Luz W.P., Pinto G., Bonifácio R., 2019	Article, Journal	Little is known about the practitioners' understanding about successful paths for DevOps adoption.	Yes, but the case study is focused at a Brazilian Government Institution	• Detail real scenarios of DevOps adoption, present a theory, a model and a case study. • Generate an adequate understanding of DevOps and assist other institutions in the path towards DevOps adoption.	DevOps adoption involves a very specific relationship between seven categories: agility, automation, collaborative culture, continuous measurement, quality assurance, resilience, sharing and transparency.

Figure A.1: Table containing general information from the first SLR

Title	Q1		Q2		Q3	
	Related		Related		Related	
E-Government For Public Values Creation: A Systematic Literature Review	Yes	• SLR, following PRISMA • Conceptual framework for public values enhancement through e-Government projects	Yes, but only DT.	Savoldelli et al.: crucial factors in e-government context could differ over the time, different cultures or countries cause problems in the adoption of DT.	No	No mention of Low-code, even process automation or even technologies for that.
Digital transformation of public services and administration: A case study of Bulgaria	Yes	• Presents Bulgaria's e-Government Model • Strategic Framework • Mobile application	Yes, but only DT.	• Existing regulatory framework did not stimulate e-government development and even hindered the use of e-artifacts – interoperability was hard. • Paperless approach: EDIMS • SEGA – State of e-government agency is mentioned.	Yes	There is mentions of technologies to adopt e-government (RegiX – Inter-Registry Exchange Environment and EDIMS) but unknown if they are low-code or EIM systems.
Towards integrated and automated management of government affairs	Yes	• System architecture: Three-tier B/S Architecture of ASP.NET • Integrated Automated Management System of Government Affairs: Approval Flow, Login, Document Management, System Management	No, at least directly	Mentions problems like lack of network security management module, like data backup, antivirus and so on, which can be interpreted as e-government problems.	Yes	• Usage of Microsoft SQL Server 2005 as relational database management system • Microsoft Visual Studio 2005 as the integrated development environment of the ASP.NET based web applications (ASP, PHP, JSP). • Utilization of flowcharts
Digital platformisation as public sector transformation strategy: A case of Ghana's paperless port	Yes	• Development of transformational affordance framework – TAF • Platformisation strategy	Yes, but only DT.	Mentions the problem that DT is well established in the private sector but lacks knowledge regarding public one, even more in less developed countries.	Yes	• Digital platform is layered. • Unknown if low-code.
An architecture for facilitating two-way G2C relationships in public service delivery	Yes	• G2C channel and architecture	Yes	• Problems with G2C communication platforms like being limited.	Yes	There is mention of Social media, G2C platforms and web or mobile applications, although no low code related. There is also mention of automation mechanisms but not in depth.
Digital Transformation? A Primer for Practitioners	No, at least directly	Although it does not mention government directly mentions other pillar which are also important not only to government but other organizations: CE – Customer Experience Transformation, BP – Business Process Transformation, BM – Business Model Transformation and OT – Organizational Transformation	Slightly	One problem presented is that DX is not only regarding automation of processes that are manual. There needs to be a transformation with focus on people, the costumers' experience.	Yes	Several open source DX technologies are mentioned Fat Free CRM, InfluxDB, D3.js, R, TensorFlow, Keras, Serverless, Apache Airflow, Activiti, PrestaShop and OpenCart, although very briefly.
Flexible artifact-driven automation of product design processes	Yes, only PA	• Automated support of business processes by information systems can be seen as state of the art for many domains • BPMN, EPCs, UML activity diagrams are state of the art for business process discovery and design	Yes	• The approaches regarding the previous question are not suited for highly creative processes, like in the field of innovative product design, requiring more flexibility.	Yes	• Use of WS-BPEL and WfMC reference model • ADEPT systems • The technology used, Oryx Project, Oryx Mashup-Framework is unknown if low-code approach or even easy to use.
IT service management automation and its impact to IT industry	Yes	• Mentions IT service management, a framework for enterprises to define, deliver, operate,... , IT services and meet business objectives. • IT service management as the implementation of IT services that meet business needs. ITIL is regarded as the best practices framework for implementing ITSM • Automation Scope Matrix	Yes	• Initial resistance is expected • Its critical to train human resources with advanced automation tools and techniques. • Requires enterprises to follow systematic approaches and processes for a successful implementation • Potential less jobs	Yes	• Mention of Basic automation, that with help from simples scripts and macros, automates repeated tasks by IT executives. Also mentioned other types of automation. • Mentions benefits of automation but unknown if low-code.
Enabling mGovernment: A framework and a case study	Yes,	• 3 stage approach: Process Transformation Framework, Mobile Content Platform, Asset Leverage Strategy – affords flexibility, scalability and mobile application • Mentions Content Management Systems • G2C, G2B (business), G2G, government's Internal Effectiveness and Efficiency (IEE) and overarching infrastructures (cross-cutting)	Yes	• One challenge is the, possible, lack of compatibility between mobile systems and existing eGovernment systems. Even harder if still using legacy systems. "The solution lies in implementing open systems using open, not proprietary standards"	Yes but not PA	• mGovernment – mobile government: use of mobile services • M-GOV project • Uses the proposed Process Transformation Framework but seems to lack features or be more specific.
The application of workflow technology in office automation system	Yes	• Mentions the Workflow Framework of COAS (Office Automation System but with Workflow), based on B/S structure.	No, directly	• There is no mention of process automation issues or challenges but office automation is mentioned several times. • Issues like large amounts of information make decisions regarding it more complex, costs associated, less flexibility...	Yes	There is mention of Microsoft Net Framework 3.0, Windows Workflow Foundation (WWF)
Need aspect approval system for procurement of components used for INDIAN satellite program	Yes	• Mentions G2E interaction, which relevant to my paper. - Mention of process flow diagram and architecture of the application used.	Yes	Mentioned earlier	Yes	Usage of 'e-NAC' approval system, a web based application software simulating workflow operations.
Deriving user requirements from business process models for automation: A case study	Yes	• Mention of framework ARIS - Entity Relationship diagram	Yes	Mentioned earlier	Yes	• Mention of PAIS, Process Aware Information Systems, developed as a workflow management (WMS) system or BPM system that executes operational processes based on formal models. • UPROM tool
Realization of administrative organs automation system based on B/S mode	Yes	• Presents a schematic topology diagram of civil air defense office automation system network • Workflow of internal examination and approval of incoming document process • B/S (Browser and Server) mode	Yes	• Complexity, heavy communication between government organs. Costs and limited trained staff are also important factors.	Yes	• Mention of Notes database and Notes server of Lotus, Domino • HTTP, HTML, URL, CGI, SSL protocols are integrated in the Domino server
Adopting DevOps in the real world: A theory, a model, and a case study	Yes	DevOps as a set of practices and cultural values that aim to reduce the barriers between development and operations teams.	Yes	Previously mentioned. Automation increases agility and reliability. It is necessary, however, to have test automation, quality checks.	Yes	Mention of ICT, information and communications technology.

Figure A.2: Table containing information regarding the research questions from the first SLR

Title	Government Related	Presents a Problem	Presents a Solution	Presents the Findings	Answers Q1	Answers Q2	Answers Q3
E-Government For Public Values Creation: A Systematic Literature Review	1	1	1	1	1	0.5	0
Digital transformation of public services and administration: A case study of Bulgaria	0.75	0	1	1	1	0.5	1
Towards integrated and automated management of government affairs	0.75	1	1	0	1	0	0.25
Digital platformisation as public sector transformation strategy: A case of Ghana's paperless port	0.75	1	1	1	1	0.5	0.25
An architecture for facilitating two-way G2C relationships in public service delivery	1	1	1	1	1	1	1
Digital Transformation? A Primer for Practitioners	0.25	1	1	0	0	0.25	0.75
Flexible artifact-driven automation of product design processes	0	1	1	1	0.5	1	0.75
IT service management automation and its impact to IT industry	0.25	1	1	1	1	1	1
Enabling mGovernment: A framework and a case study	0.75	1	1	1	1	1	0.5
The application of workflow technology in office automation system	0.75	1	1	0	1	0	1
Need aspect approval system for procurement of components used for INDIAN satellite program	0.75	1	1	1	1	1	1
Deriving user requirements from business process models for automation: A case study	0.75	1	1	1	1	1	1
Realization of administrative organs automation system based on B/S mode	1	1	1	1	1	1	1
Adopting DevOps in the real world: A theory, a model, and a case study	0.75	1	1	1	1	1	1

Figure A.3: Table containing the results/score and metrics evaluated in the first SLR

Title	Score	Maximum Score	Percentage score
E-Government For Public Values Creation: A Systematic Literature Review	5.5	7	78.6%
Digital transformation of public services and administration: A case study of Bulgaria	5.25	7	75%
Towards integrated and automated management of government affairs	4	7	57.1%
Digital platformisation as public sector transformation strategy: A case of Ghana's paperless port	5.5	7	78.6%
An architecture for facilitating two-way G2C relationships in public service delivery	7	7	100%
Digital Transformation? A Primer for Practitioners	3.25	7	46.4%
Flexible artifact-driven automation of product design processes	5.25	7	75%
IT service management automation and its impact to IT industry	6.25	7	89.3%
Enabling mGovernment: A framework and a case study	6.5	7	92.9%
The application of workflow technology in office automation system	5.5	7	78.6%
Need aspect approval system for procurement of components used for INDIAN satellite program	5.25	7	75%
Deriving user requirements from business process models for automation: A case study	4	7	57.1%
Realization of administrative organs automation system based on B/S mode	5.5	7	78.6%
Adopting DevOps in the real world: A theory, a model, and a case study	7	7	100%

Figure A.4: Scores and percentages regarding the results obtained from the first SLR

Paper Name	Modules
Is e-government serving companies or vice-versa?	Portal (including translation), Documentation.
Deriving user requirements from business process models for automation: A case study	Eletronic Document Management System, Web services (such as Payment), Legal Entity System, Central Civil Rights System, Entities, Communication, Roles.
Ad-hoc business process management in enterprises as expert communities	Business Process Models, Roles, Services.
A research on the enterprises information management based on e-commerce	Information Support Layer, Management Support (link between Databases and Application Layer), Application Layer, Services (such as Payment and present Visitors Info), Databases, Communication, Legality.
The changing face of government: The trends and a solution architecture for citizen services	Channel communication (web/mobile) to G2C and G2E, Portal, Content Manager, Transactional Service, BP Choreography,, Database, Web Services (SMS), Agency System, Social Media component. Entities.
Discovering business models for software process management: An approach for integrating time and resource perspectives from Legacy information Systems	Databases, BPMN Portal to use that Information (mentions the need to transform data which is unnecessary by using IMS – Information Management Systems).
Mobile supported and process enabled electronic document management system for local municipalities	Document Management, e-Government Portals, Notification Process, Client Layer, Server Layer (WebServer, Service Server, Database Server), legality, Roles, Entities.
The e-Government interoperability through Enterprise Architecture in Indian perspective	Portal (GEAF) uses Internet to Citizen, Online Transactions, Database, communication through NSDG.
Supporting legal requirements in the design of public processes	Communication, Legal Requirements Module, Roles, Entities.
Business process management: A holistic management approach	Business Process Models, Regulations, Communication, Documentation/Content, Roles, Permissions.
Modeling of Information System for Licensing and Extension of Special Hajj Pilgrimage Organizer	Documentation, Network connection between Government Departments, Portal, Web Services (such as add documents and track status), BPMN.
Enterprise Content Management and the Records Continuum Model as strategies for long-term preservation of digital information	Documents, records, communication, requirements
A dynamic-capabilities view of local electronic government: Lessons from two successful cases	Web Services (such as Payment), Portal that contains E-services (Procedures/Transactions), Communication, Monitoring, Mobile.
Proposal for application of data science methods in E-Government: A Case-study about the application of available techniques for performance measurement with the help of data science	BPMN, Databases, Portal, Roles, (Web) Services, Documentation, Communication, Roles.
Towards Setting Up a Collaborative Environment to Support Collaborative Business Processes and Services with Social Interactions	User application, BPMS layer, Integration Layer, Social Media, (Web) Services (such as Notifications, Login, Creation and Viewing of process), Documentation, Web Site portal, BPMN.
Service Oriented Design for Indonesian E-Government System Using SOA	Processes, Core Services (such as payment), Portal, Database, Documents (Files), Communication, Entities, .
Intelligent information management with digitization workflow	Security methods (Face Recognition), Documents/Records Databases, Communication, SaaS user interface, Regulatory Compliance, PaaS Process service and back office service, Portal for Content Distribution (such as Internet or E-mail).
Codezin: braving the startup storm	Web Portals, BPM Models, Communication (with users through E-mail, social media), (Web) Services (such as Notification).
MARREG - MARriage REGistration [for a better cause]	Portal, Web Services (such as Payment and Notifications), Communication (as E-mails), Document gestion, Databases, Compliances.

Figure A.5: Modules present in each paper from the second SLR

Paper Name	Modules								
	Web Pages	Documentation	Business Process Models	Communication	Web Services	Entities	Roles	Security/ Permissions	Citizen Application
Is e-government serving companies or vice-versa?	X	X							X
Deriving user requirements from business process models for automation: A case study		X	X	X	X	X	X		X
Ad-hoc business process management in enterprises as expert communities		X	X		X		X		
A research on the enterprises information management based on e-commerce	X	X		X	X	X			X
The changing face of government: The trends and a solution architecture for citizen services	X	X		X	X	X			X
Discovering business models for software process management: An approach for integrating time and resource perspectives from Legacy Information Systems	X	X	X						
Mobile supported and process enabled electronic document management system for local municipalities	X	X		X	X	X	X		X
The e-Government interoperability through Enterprise Architecture in Indian perspective	X	X		X	X				X
Supporting legal requirements in the design of public processes				X		X	X		
Business process management: A holistic management approach		X	X	X			X	X	
Modeling of Information System for Licensing and Extension of Special Hajj Pilgrimage Organizer	X	X	X	X	X				
Enterprise Content Management and the Records Continuum Model as strategies for long-term preservation of digital information		X		X					X
A dynamic-capabilities view of local electronic government: Lessons from two successful cases	X			X	X		X		X
Proposal for application of data science methods in E-Government: A Case-study about the application of available techniques for performance measurement with the help of data science	X	X	X	X			X		X
Towards Setting Up a Collaborative Environment to Support Collaborative Business Processes and Services with Social Interactions	X	X	X	X	X				X
Service Oriented Design for Indonesian E-Government System Using SOA	X	X		X	X	X			X
Intelligent information management with digitization workflow	X	X		X	X				
Codezin: braving the startup storm	X		X	X	X				
MARREG - MARriage REGistration [for a better cause]	X	X		X	X				X

Figure A.6: Table presenting if each paper, from the second SLR, contains the main modules identified

	Score	Score Max
Is e-government serving companies or vice-versa?	3	9
Deriving user requirements from business process models for automation: A case study	7	9
Ad-hoc business process management in enterprises as expert communities	4	9
A research on the enterprises information management based on e-commerce	6	9
The changing face of government: The trends and a solution architecture for citizen services	6	9
Discovering business models for software process management: An approach for integrating time and resource perspectives from Legacy Information Systems	3	9
Mobile supported and process enabled electronic document management system for local municipalities	7	9
The e-Government interoperability through Enterprise Architecture in Indian perspective	5	9
Supporting legal requirements in the design of public processes	3	9
Business process management: A holistic management approach	5	9
Modeling of Information System for Licensing and Extension of Special Hajj Pilgrimage Organizer	5	9
Enterprise Content Management and the Records Continuum Model as strategies for long-term preservation of digital information	3	9
A dynamic-capabilities view of local electronic government: Lessons from two successful cases	5	9
Proposal for application of data science methods in E-Government: A Case-study about the application of available techniques for performance measurement with the help of data science	6	9
Towards Setting Up a Collaborative Environment to Support Collaborative Business Processes and Services with Social Interactions	6	9
Service Oriented Design for Indonesian E-Government System Using SOA	6	9
Intelligent information management with digitization workflow	4	9
Codezin: braving the startup storm	4	9
MARREG - MARriage REGistration [for a better cause]	5	9

Figure A.7: Scores from each paper, from the second SLR