DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2022

Title: Proposed User Interface for an Archive of Archives

Author: Callum Fraser

Project Abbreviation: ARCH2

Supervisor(s): Hussein Suleman

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	**80**

User Interface for an Archive of Archives

Callum Fraser
Computer Science
University of Cape Town
Cape Town, South Africa
FRSCAL004@myuct.ac.za

ABSTRACT

Digital archives are crucial in preserving digital items for use in the future. However, digital archives themselves can be at risk of disappearing, particularly in low resource environments. This can lead to important information being lost without a trace. To address this problem experimental research is being conducted around preserving archives themselves in a new type of archiving system, known as an archive of archives. This paper contributes to the current research into an archive of archives by proposing a user interface for one. Three core features were implemented to distinguish an archive of archives from a normal archive. The first feature displayed the website of stored archives in different ways, while the second provided a timeline to show how each archive changed over time. Finally, the last feature involved a new archiving administration system to manage the adding of archives, and retrieval of their data. All these features were evaluated on their usability through user interviews. Thematic analysis on the results showed that the presented features were all easy to use and easy to learn, which suggests that the proposed features will be suitable for future work into an archive of archives.

KEYWORDS

Archiving, Archives, User Interface, UI, Usability

1 Introduction and Background

Archives are used for the long-term storage and preservation of information. Digital archives have become very important to modern society due to the importance of storing historically significant digital information and research for the future regardless of the rapid change of technology. While Digital archives have many benefits over traditional archives, they must overcome very different challenges and are constantly at risk of excluding certain groups of people or facing their own demise [1, 2]. A key area of research is in trying to establish the most effective way to digitally preserve knowledge and to do so for a period at least comparable to existing preservation methods. Modern computing is very new, and the dramatic speed of change creates uncertainty around the digital preservation process and which methods will last the test of time [3].

Both a digital archive and a digital library (DL) must provide a way of adding, organising, cataloguing, and easily retrieving large amounts of information through a user interface [4]. Archives are separate to digital libraries in their explicit mission of preservation, which typically involves the use of well-defined techniques to prevent the original artifact from deteriorating over time [3]. Digital preservation encompasses the activities required for maintaining the computer bytes required to effectively reproduce the original content in an accessible way that can continue long into the future and with changing technology [5]. However, concerns still exist whether true digital preservation can be a reality [1]. This puts digital archives at the forefront of information research.

Over time, more unstructured digital information is being generated, which must be organised and stored, with the hope that significant content is preserved [6]. However, a major issue that has arisen with network computing is that information moves location or disappears without a trace and information that was assumed to be permanent or of significant importance is lost forever [6]. Digital archives have attempted to address issues around the disappearance of digital information by preserving them in catalogued collections. However, there is still a concern of archives themselves disappearing with all their preserved information. Digital archives do not only face concerns around the underlying technology but also on the organisations maintaining them, as they have great liberty in the methods and accuracy of the preservation processes applied [1]. This results in the existence of the archive depending on continued funding and commitment of its maintaining organisation. It is archives of low-resource environments that have major concerns around the preservation of data as they are more likely to lack funding or the correctly skilled people to maintain them.

Archives need to create trust in their ability to preserve information, no matter the funding afforded to the managing organisations [7]. Thus, the possibility arises for establishing research around the creation of a new breed of digital archives that preserve other digital archives in the hope of creating trust, versioning, and long-term persistence of data through the conservation of other archives. This paper forms part of broader research to establish the first archive of archives (Arch2), which attempts to protect and share already persisted digital archives specifically to be more inclusive for low resource groups of people. Such an archive of archives would involve three core components, a scraper that extracts the data of a current archive, an ingesting component that persists the data, and a user interface that displays the contents. See Part A.1 of the Appendix for the proposed architecture. This paper proposes the user interface (UI) for such an application that attempts to differentiate a higher-dimensional archiving system from its standard counterparts. For this research, an experimental user interface prototype for an archive of archives was developed as an extension to the Simple DL toolkit.

2 Related Work

While no archive of archives exists there is previous work that attempts to tackle some of the same problems or provides the foundation that this work, and the larger research around an archive of archives, can extend. Previous research has been conducted looking into the merits of new methods of preservation and making use of client-side technologies to enhance archives. Additionally, tools exist with core features that could be useful in creating a higher dimensional archive or one that supports low resource environments, such as by providing offline support and easy transportability.

2.1 Web Archiving

Web Archiving has become a major part of digital archiving due to the vast amount of information on the Internet [8]. The Internet Archive is one of the largest and most well-established digital archives in the world and effectively stores websites. It is a massive 25-year-old Web archive that preserves petabytes of data while being managed by a tiny team of under 10 people [9]. A core feature of the Internet Archive is its Way Back Machine that provides an access tool with the ability to retrieve stored Web pages through URL search [10]. The Internet Archive's Way Back Machine presents data storage, access, and versioning. The versioning and visualising used by this archive suggests a model for how archives could be displayed when persisted and shows how persisted information can be useful even with the original site still online. However, a downside of the Internet Archive is its lack of client-side capabilities, which could be addressed with features like local browsing or easy redistribution of content.

2.2 Client-side Archiving

Client-side archiving has emerged as a trend, as the increased processing capacity of everyday computers reduces the dependence on centralised servers [2]. New research is being conducted to try to distribute archives to clients with less dependence on networks [9, 10]. Multiple approaches have been taken, including work to develop a system that focuses on reproducing and storing the functionality of an archived web application instead of only preserving its data [11]. The ServiceWorker web API has been suggested as another way of maintaining online or linked functionality without the need for the original hosed environment [12]. Another new development focuses on a search solution to address concerns around unreliable network access by running client-side [9]. Additional research has shown it is possible for clients to perform more of the general processing in the browser using the Ajax development techniques [13]. Simple DL is an experimental toolkit that focuses on client-side archiving techniques to reduce complexity and better support low resource environments [14]. It provides a means of pre-generating system independent digital libraries from assets presented in flat file formats, without the need for special installed software dependencies. It uses Extensive Markup Language (XML) files to store metadata in simple files that can be used to generate static sites, which are viewable on any browser without requiring a database. This toolkit is well suited for the purposes of archiving archives as it can facilitate preservation without active management, it reduces the need for additional software installation, and can be easily extended [14]. For these reasons Simple DL will be used as a key part of this project. However, Simple DL does not currently provide a model for addressing the concern of archives themselves disappearing.

2.3 Extending Digital Preservation

More diverse types of data are being persisted with new approaches for guaranteeing the continuous development and support for the underlying technology [2]. Fedora is an older archiving system that was designed for extensibility using a service-orientated architecture (SOA) [15]. Microservices have been suggested as a modern alternative that promotes customisation and decreases risks of failure by implementing independent applications [16]. The microservices architecture is generally simpler to manage than a SOA due to having multiple independent services that work together rather than a single overarching scope to connect the components [17]. Research has also been conducted to look at ways of preserving data in low resource environments, which allow preservation without network access, through the use of simple files that have wider support and allow for flexibility [18]. Additionally, it has been suggested that providing methods of distribution can aid in the preservation process, particularly when it can occur offline [19]. GitHub encompasses many novel preservation principles by storing and allowing easy distribution of millions of repositories of versioned code that are created and tracked with the Git software [20]. Like an archive, this versioned code is intended to be stored safely for extended periods of time and provide backups if needed. The copying, modifying, and sharing of stored repositories is a key feature of GitHub, including the creation of new repositories from previous ones [21]. This forking feature allows users to have their own independent versions of other repositories. Even though GitHub stores code repositories rather than digital archives, its higher order structure matches that of an archive of archives, which aided this research by highlighting potential features and their graphical implementations.

3 Research Aim

3.1 Research Description

The aim of this research is to determine whether an experimental user interface (UI) can be built, which allows users and administrators to effectively interact with a complex higher order digital archive that persists the data of other archives. In doing so this contributes to a larger research project that attempts to establish the general feasibility of archiving archives. Specifically, this archive UI will be made up of the data of other online archives that have been stored and must be displayed. Given that there is no previous implementations or research that creates an archive of archives, this work presents a suggestion of what such a system looks like. Part of this is establishing a way of showing how an Arch2 is different to a traditional archive. Additionally, it must convey its higher-order nature to potential users and allow them to distinguish an original archive from the preserved copy. Since the research is only part of broader work, it does not attempt to answer anything about the overall usefulness of an archive of archives but narrowly focuses on whether the suggested UI would be usable as part of the greater system.

3.2 Research Question

To what degree does the presented model for the user interface (UI) for an archive of archives (Arch2) allow consumer and admin users to effectively interact with it, based on their usability experience, despite its differences to standard archives that result from its higher order structure?

3.3 Evaluation

Evaluation of the research occurred around the usability and general experience of interaction with the UI. The user evaluation of the system was broken into two core parts: the general interface for consumers of the archive and the admin interface for archive managers. These two components were tested with different participants based on their prior experience in managing digital archives.

4 System Development and Implementation

The system was developed and implemented as an extension of the Simple DL toolkit. As this research only focused on one component of what is required to create a production archive of archives it assumed the existence of certain features of an Arch2 backend that can scrape the data of publicly accessible online archives and process them for preservation. Thus, the UI simulates these features where they are not yet implemented in the default Simple DL toolkit.

4.1 Key Features

There are three major features proposed to differentiate the Arch2 UI from that of other archives. The first feature is to display what the stored digital archive originally looked like so that it is clear it is a full archive and not just a digital record. The second feature is to show the evolution of the archive over time, giving people insight into how the archive changes even when certain individual records stay the same. The third feature is to provide an interface for administrators to add and manage full archives beyond the standard management of archive records, specifically, using scraping of an archive's website as the means of data retrieval. This included getting and showing details of the scraping archives.

4.2 Simple DL Extension

The proposed UI is built as an extension of Simple DL. By utilising the tool's core features, it made it easier to solely focus on the key features of an Arch2 and not rebuild common archiving elements. The original plan was to use Simple DL as the backend with a UI built using the Dart language and Flutter application framework. While this would make the most of the toolkit's CSV importing and XML indexing features, it made more sense to also include its static generation features, which convert XML to static HTML (Hyper Text Markup Language) Web pages. This allowed the built-in search of Simple DL to work straight out of the box and avoided concerns around mixing and matching static and dynamic pages that may not comply with Simple DL's design principles. Additionally, this would allow the system to already contain an administration and standard user interface that has already been used in live sites for hosting standard archives and can more easily integrate into future work around the toolkit. In the end all the key features were implemented in the Extensible Stylesheet Language (XSL) that is already used in Simple DL.

4.3 Software Development Process

The development process chosen for this work did not follow a traditional process but used the principles of multiple processes to get an effective one specifically for this work. The sequential phase-based approach of the Waterfall Model was combined with the refining of the Incremental Software development model. The overall work was split into sequential phases like that of the Waterfall method [22]. However, instead of having the typical requirements analysis, designing, coding, testing, and maintenance, the phases for this project started with a slightly different approach with a literature review and design proposition rather than a requirements analysis. Additionally, the coding phase of Waterfall Method was done slightly differently by being broken into multiple sub-phases that could make use of iteration in a similar way to the Incremental Model [22]. Coding was broken into an initial demonstration of the proposed concept followed by independent development phases for each key feature. Each feature was refined and iterated on while developing the next.

4.4 Development Environment

A specific development environment was selected to make the coding process simpler. Since no modifications were required of the core Simple DL software toolkit, this code was put into a Docker Image, which allowed the creation of a standardised container that was easily transportable and provided everything to run Simple DL on any operating system. The parts of the toolkit being extended were copied into the running Docker container in real-time, which allowed that only the parts being changed were tracked by git in their own instance. This simplifies the steps needed for adoption by future researchers, as they can run the service with the exact same environment and instance of Simple DL with only the necessary changes tracked.

4.4.1 Code Layout

Simple DL makes use of a single XSL transformation file, located in the config sub-folder of the data folder, which generates all

XML into HTML for displaying. All extensions made to the transform were added to separate files that were named based on their functions and then imported into the original transform so that future developers can see which parts have been modified. Additionally, new pages that did not already exist in Simple DL were added as XML files to be transformed. These pages were located under the website sub-folder of the data folder.

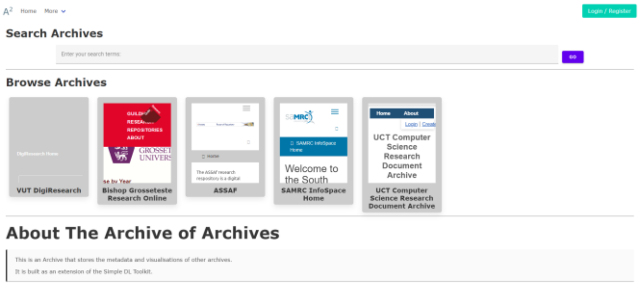
4.5 Implementation

Most of the standard styling and transformations used by Simple DL were kept, however additional CSS was written for the archive viewer and new styles were added from the Bulma CSS library, which was downloaded and stored in the styles folder. The CSS framework was chosen for being a simple open-source framework that did not require JavaScript and had minimal interference with Simple DL's original CSS. Additionally, AlpineJS was statically added to provide JavaScript extensions. It was chosen for being a small, minimalistic framework that would not add too much complexity and could easily integrate with the XSL transformations.

The standard multi-page website system of Simple DL was maintained, with a main page that allows scrolling between collections that was extended to be exclusively archives. When clicked, archives open a new page with their content. The typical collection page used in Simple DL was extended to be an archive page, representing a single stored archive. Archive pages allow the viewing, downloading, and previewing of the archive, as well as scrolling through the metadata of their individual items. Archive pages also provide general archive information and allow access to the archive version timeline.

4.5.1 Archive Visualisation

The Archive Visualisation focused on displaying what the stored digital archive originally looked like so that it could clearly be seen that it is a full archive and not just a digital record. This was built using an HTML iframe tag to visualise the Web page of an archive that has been stored in a zip folder. It used Simple DL's built-in view function to display the file while zipped. This visualisation part was added to the archive_viewer_transform.xsl file. Archive visualisations were added to the home page of Simple DL so that stored archives would display previews of their websites instead of icons that typical collections would display, this can be seen in Figure 1 below. Next, the standard transform for generating pages for collections of content was extended so that a larger, scrollable preview of the archive it represents would be the first thing users would see when opening a particular archive. This extension can be seen in Figure 2 below. Finally, a view button was added so that users could open a page that exclusively shows the archive preview on a full page. The full view was wrapped by a border so that users would know that they are still on the stored page and not the original archive site, as seen in Figure 3. The details of page are passed in the URL to the single viewer page.



Search Archives

Enter your search terms:

Browse Archives

DigiResearch Home

VUT DigiResearch

Bishop Grosseteste Research Online

The ASSAF research respository is a digital

ASSAF

Welcome to the South

SAMRC InfoSpace Home

UCT Computer Science Research Document Archive

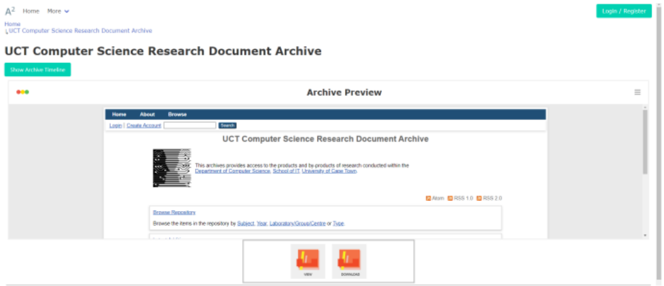
UCT Computer Science Research Document Archive

About The Archive of Archives

This is an Archive that stores the metadata and visualisations of other archives.

It is built as an extension of the Simple DL Toolkit.

Figure 1: The extended home page showing example archives with previews of their websites as covers.



Home
UCT Computer Science Research Document Archive

UCT Computer Science Research Document Archive

Archive Preview

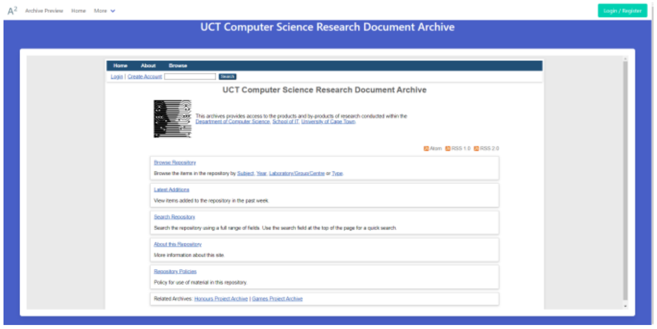
UCT Computer Science Research Document Archive

This archives provides access to the products and by-products of research conducted within the Department of Computer Science, School of IT, University of Cape Town.

Browse Repository

Browse the items in the repository by Subject, Year, Laboratory/Group/Centre or Type.

Figure 2: The Archive Preview used at the top of the archive page.



UCT Computer Science Research Document Archive

UCT Computer Science Research Document Archive

This archives provides access to the products and by-products of research conducted within the Department of Computer Science, School of IT, University of Cape Town.

Browse Repository

Browse the items in the repository by Subject, Year, Laboratory/Group/Centre or Type.

Latest Additions

View items added to the repository in the past week.

Search Repository

Search the repository using a full range of fields. Use the search field at the top of the page for a quick search.

About this Repository

More information about this site.

Repository Policies

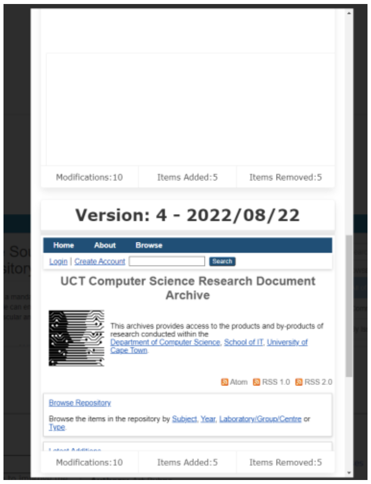
Policy for use of material in this repository.

Related Archives: Honours Project Archive | Games Project Archive

Figure 3: The Archive View page providing a full screen preview of an archive's original website.

4.5.2 Timeline/Versioning

The Timeline or Versioning feature shows users the evolution of an archive over time. It displays a popup when a button at the top of an archives page is clicked. A user can scroll down the popup to see a preview of the archive after each scrape; by clicking on the version the user is taken to a full-page preview of the version. An example of the timeline is shown below in Figure 4, where the user has already opened it and scrolled down to the fourth version. JavaScript is used to display the version timeline with all the custom scripts and XSL added to the archive_versioning_transform.xsl file. To implement the versioning details a JSON file was written with mock versioning data for stored archives.



Modifications:10 | Items Added:5 | Items Removed:5

Version: 4 - 2022/08/22

UCT Computer Science Research Document Archive

This archives provides access to the products and by-products of research conducted within the Department of Computer Science, School of IT, University of Cape Town.

Browse Repository

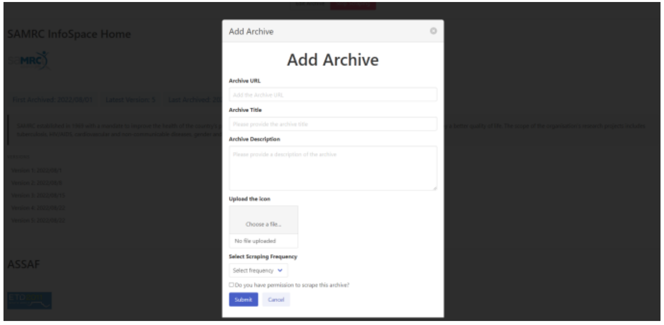
Browse the items in the repository by Subject, Year, Laboratory/Group/Centre or Type.

Modifications:10 | Items Added:5 | Items Removed:5

Figure 4: The implemented archive timeline, which can be clicked on to display a full page of a certain version.

4.5.3 Archive Manager

The Archive Manager was built as a unique administration feature for adding new archives and managing the scraping of stored ones. Simple DL already had an authentication system and admin page. The admin page was modified in the admin_transform.xsl file so that admins could see a link to a new archive manager page. This page was generated from extensions made in the archive_manager_transform.xsl file. The archive manager page would display all the archives stored with data about their scraping and would allow the admin to add a new archive by clicking a button. Once an admin fills out the add-archive form, which is shown below in Figure 5, example feedback is given emulating the scraping and ingesting process with a status bar at the top of the page. Once finished, the administrator can download a report of the ingestion or make edits to the stored archive.



SAMRC InfoSpace Home

First Archived: 2022/08/01 | Latest Version: 5

Version 1: 2022/08/1
Version 2: 2022/08/8
Version 3: 2022/08/15
Version 4: 2022/08/22
Version 5: 2022/08/22

ASSAF

Add Archive

Add Archive

Archive URL: Add the Archive URL

Archive Title: Please provide the archive title

Archive Description: Please provide a description of the archive

Upload the icon: Choose a file... No file uploaded

Select Scraping Frequency: Select frequency

□ Do you have permission to scrape this archive?

Figure 5: The Add Archive Popup on the archive manager page, with archive data in the background.

5 Experiment Design and Execution

Qualitative interview-based experiments were created to test the research question around the usability of the proposed system. A Usefulness, Satisfaction and Ease of Use (USE) questionnaire was converted to open-ended questions for the interview format. Interview results could then be thematically analysed to answer the research question.

5.1 Experimental Design

The initial plan was to perform online surveys that would be self-administered by participants. However, this was pivoted to in-person interviews on the recommendation that they would present better results by allowing participants to have more freedom in their feedback and a better understanding of the experimental system through interaction with a researcher who can answer questions and react to participant insights. Furthermore, by having in-person interviews, participants all engaged with the system and trivial answers were avoided, as might have occurred with an anonymous survey.

The interview was designed to require users to complete tasks to check the intuition, usability, and experience of interaction with a prototype version of the system. Once participants completed the assigned tasks, an interview was conducted so that each user provided their perception of the usability, satisfaction, and learnability of the system as well as additional feedback. This format sought to find out whether the proposed experimental system answered the research question.

5.2 Research Methodology

The USE questionnaire was chosen based on its popularity and proven evidence of reliability and validity [23]. It attempts to treat the dimensions of usability as dependent variables, which is desirable in an experimental context [24]. It focuses on simplicity and uses well established standard psychometric techniques [23]. While the traditional test makes use of a seven-point Likert rating scale, the questionnaire was modified so that the relevant parts were chosen and presented as open-ended questions more suitable for an interview format. Importantly, the usefulness section of the USE questionnaire was removed entirely, leaving only the Ease of Use, Ease of Learning, and Satisfaction sections. This was done due to the experimental nature of the system, which in its prototype stage cannot be tested for usefulness since no current Arch2 users exist. Separate USE questions were customised for the admin and non-admin sections of the evaluation. Additional open-ended questions were also created for both sections that attempted to find the most positive, negative, as well as enjoyable features of the application. Lastly, a question seeking general comments was asked for in both parts of the system. In total there were eighteen questions for experienced users and twenty-six for non-experienced users. Non-experienced users were also asked a few background questions to determine the participant's knowledge of archives and user interfaces in case it influenced their experience.

5.3 Participants

Two different types of users were recruited based on their experience with archives; by using these groupings a better triangulation of the overall user experience could be assessed based on their different perspectives [25]. Experienced archive users were used to test the admin interface as they would have knowledge and insight into how an archive management or similar system would be used. On the other hand, inexperienced archive users were used to test the general interface as no previous experience with an archive should be required for a highly usable Arch2 UI. Experienced archive users could evaluate all sections of the UI, however in practice this was only done if the participant had more available time.

5.3.1 Participant Sampling

Convenience Sampling was the overarching method used to select participants. Convenience sampling provided an easy and efficient way of getting participants, which was important given the short time to conduct the research [26]. Participants who were willing to volunteer and eligible were chosen. Inexperienced archive users could be recruited from the general populous with little discretion. Due to the intimate nature of interviews and the requirement of getting specialised users' all participants were recruited through direct contact. Recruiting was mostly conducted through in-person engagement, as it was easier to book a time without a difficult online negotiation, which increased the likelihood of participation. Participants with a close relationship to the researcher were avoided to reduce the chance of any bias in feedback. Naturally a degree of Snowball Sampling occurred as participants recommended other persons who might be willing volunteer. Snowball Sampling was practical and allowed less time to build trust with participants [27]. Normally, there is a risk around the quality of referrals when the Snowball Sampling occurs, but this was mitigated because it was only used for recruiting inexperienced archive users.

Data Saturation was used to determine the sample size, which is the standard practice for conducting qualitative research [26]. This required estimating and then validating the number of interviews required for no new information to be observed. The clear scope of the research meant that a smaller sample could be used to reach data saturation [26]. Different research has estimated the sample size of usability testing at around five and ten users respectively, which have made these numbers generally accepted standards [28, 29]. Based off these different factors, the end sample size of participants for this research was five experienced archive user interviews and nine inexperienced archive user interviews, with two experienced archive users conducting interviews on all features.

5.3.2 Experienced Archive Users

Experienced archive users all either had interest in, or experience managing archives or similar software systems. These people were recruited for being members of the UCT Digital Library research group, or for being professionals who had previously worked with specialised digital library systems.

5.3.3 Inexperienced Archive Users

Nearly all inexperienced archive users were UCT students who were directly contacted and could recommend others. These students were young persons who had been regularly exposed to user interfaces, which provides them with experience for understanding usability. Additionally, these students represent the future generation of researchers and archivists/managers who may utilise an archive of archives.

5.4 Experimental Ethics

Ethical clearance was obtained by the University of Cape Town (UCT) Faculty of Science Ethics Committee to go ahead with evaluation and approval from the UCT Department of Student Affairs was received to conduct research with UCT students. Participants were provided with a consent form and written consent was gathered for all participants engaging in an interview. Research was conducted ethically by ensuring that participants had their anonymity protected and consented to all evaluations conducted. No foreseeable risks existed for participants and no strenuous activities were required.

5.5 Experiment Execution

Participants were given the same brief and basic tasks to interact with the system so that each would have the same initial understanding. All experimentation was done on the same 15-inch laptop. The brief gave an overview of what an archive of archives is and where this research fits into it. The instructions provided a few simple steps that would highlight the key features of the Arch2 UI. Non-experienced participants had six short tasks that would show them the different archive visualisations and have them use the archive timeline. Experienced participants had ten steps that would allow them to interact with the core parts of the admin archive manager interface, including adding a new archive. Participants were allowed to do additional exploring of the system by themselves and could navigate the application during the interview, if they required it.

Standardised questions were used for all interviews so that all participants would get the same prompts. All interviews were conducted in a neutral environment to try to promote the sense that participants could freely express their opinions without fear of offending the interviewer. The anonymity of the results hopefully contributed to participants feeling more open to sharing their opinions. Interviews were conducted in a way that would leave the participants time to respond after each question, with further prompts and follow up questions based on their responses. This allowed participants to provide reasons behind their responses and to seek clarity on any misunderstandings that might arise from an interpretation of the prewritten question. Typical interviews took between 15-20 minutes.

6 Findings

Thematic analysis was used to categorise the qualitative data gathered from the interviews and in doing so identify meaning. This process was driven by the research question to avoid categorising every possible theme in the data and to ensure focus on the key aspects that relate to the current research.

6.1 Findings Breakdown

Eleven interviews were conducted on the non-admin features, which was made up of nine inexperienced archive users and two experienced archive users. Five experienced archive users provided feedback on the admin features. The feedback from the interviews was positive, with participants finding the features easy to learn and easy to use. The only general concern from users was around the aesthetics of the system, which affected perceptions of satisfaction. However, styling was not the priority of the research and thus negative feedback was not unexpected given the focus on functionality and the short time of development. Additionally, a general pattern was observed where participants would find a single focus for their criticism and praise throughout an entire interview, rather than providing diverse comments with different shortcomings and strengths.

6.2 Thematic Analysis

A thematic analysis was conducted on the findings because of the subjective nature of interpreting the feedback of participants experiences of the UI. This was aligned with the qualitative research question that sought to find the degree of usability for the proposed UI.

6.2.1 Thematic Analysis Process

Thematic analysis is an analysis process that identifies patterns in qualitative data to find meaning [30]. It is particularly appropriate for understanding the experiences of participants, as is the case with this research [31]. Since the explicit opinions of participants were sought based on questions already grouped by USE sections, a combination of the deductive and semantic approaches to thematic analysis was taken. This meant focusing on themes that relate to the underlying USE questionnaire and taking responses on face value rather than attempting to find underlying meaning in participants' opinions [32]. A reflexive type of thematic analysis was used to adapt the overarching groupings to themes that arose from the participants' feedback. A reflexivity journal was kept while analysing the results so that data coding could be tracked throughout the process to increase the reliability of the results by ensuring analysis was systematic and consistent. The USE themes (ease of use, ease of learning, satisfaction) were used as the starting codes in the journal. Additional themes were added, and some removed, over time. After developing them, all themes were reviewed to ensure accuracy and comprehensiveness. All themes were finalised with labelling to fully describe them and ensure they aligned with the research question.

6.2.2 Standard User Interface

The following themes were extracted from participant feedback around the archive visualisation and timeline features, which came from eleven interviews.

6.2.2.1 Theme 1: Ease of Use

The theme of ease-of-use describes the comments participants made about the simplicity, easiness, and friendliness of interacting with the UI. It was chosen because of the positive feedback participants gave on questions related to the USE theme and the wording participants chose to answer other questions. For instance, all participants agreed that the application was easy to use, with only a two expressing concerns. Nine participants asserted that they could use the application without written instructions with one participant giving the caveat that they would need to know what they were doing. Only two participants said that they would need instructions to interact with the UI. Similarly, everyone interviewed agreed it was simple to use the timeline feature, with only one participant commenting that it was hard to get to it, while another said that better spacing could be used with the styling of it. Further ease-of-use feedback was received with all besides one participant saying that they could easily recover from mistakes while using the application. The one participant in disagreement mentioned that it may be hard to recover from a mistake because, “navigating inside an archive's displayed preview/view [can be unrecoverable], when it's displayed in the window since the back button doesn't work.” This referenced the fact that there was no back button inside the navigation of a stored archive's preview, which may require a user to refresh the page to get back to the preview archive's home screen unless the stored website has its own navigation mechanism. This could be a feature that is added later. Regardless of a few concerns, four participants agreed that the most positive aspect of the application was the ease-of-use, which was the most for a single concept, with one participant saying, “You could be a first-time user and still figure out how to navigate it easily”.

6.2.2.2 Theme 2: Aesthetics

The theme of aesthetics was chosen based on criticism of the styling of the UI and the common phrasing by participants to explicitly use the word “aesthetics” in their feedback. On the topic of being user-friendly, seven participants explicitly said it was, but a common theme emerged about the aesthetics of the design as a detractor among the other participants. One participant explicitly stated that “it's not aesthetically pleasing”. Another stated that “it could have a more modern aesthetics look”. A third stated that, “I think it needs more aesthetics”. The frequency of the word “aesthetics” emphasised the need for the theme. When asked about the satisfaction of the interface aesthetics came up. This caused satisfaction to have mixed results even though five participants found all the features satisfying without any concerns. One participant noted that they found it satisfying by saying, “[I] clicked on the archive based on how the [icon] preview looked interesting.” While another stated that they did not find the archive visualisations satisfying because they found the initial archive icons too small and they did not like the colour styling saying, “the cards are too small for the [archive preview] icons” and “[I am] not a fan of the colour palette.” At least two participants commented on how to make the design better, one said, “more colours and labels.” Another stated that, “the view and download buttons should be up at the view timeline button.” The topic of aesthetics was collectively seen as the most negative part of the application when participants were asked to name one, with ten of the elven participants commenting on the styling. Additionally, eight participants listed design improvements as the single most needed changes, however there was disagreement on what these would be. For example, when commenting on the design of the preview one participant suggested "putting the preview at the bottom of the archive page", which another contradicted by saying "pull things up and make the preview a bit bigger." Additionally, one participant commented that they enjoyed the versioning feature the least saying, "it was a bit confusing to use". This contradicted the views of other participants as the versioning feature was perceived as the most enjoyed feature by four of them, with one saying that "[I] like the timeline and being able to scroll through the different versions." Importantly, even though participants found the application to be lacking in design most agreed that this did not affect its functional purpose. Nearly all participants were satisfied by the how the core features met their expectations. All eleven participants agreed that the versioning worked as expected and ten agreed that archive viewing worked as expected. However, one participant stated that they expected the view function to "open a new tab" "like when you click on a pdf". Thus, despite design issues the overall satisfaction was not badly perceived.

6.2.2.3 Theme 3: Ease of Learning

Ease-of-learning describes all the feedback on the lack of difficulty and speed required to learn and remember how to use the application. It emerged as a theme due to the popularity of positive responses to questions on it. For example, all participants agreed it was easy to learn to navigate between archives, with only one participant stating that instructions would help. Similarly, eight participants stated that it was easy to learn to change the version of an archive, with the others indicating that instructions would be required for first time usage. One participant said that "It would take a few times to get the hang of it and fully understand it." Only two participants found the overall application difficult to use rather than easy, one stated that "it was very difficult to understand the complexity of the app" and the other saying that it was, "confusing having two buttons with the same image." For questions on ease of memorability only one participant said that the UI would not be, but they blamed themselves saying, "[I am] technologically challenged". On the other hand, all agreed that they could quickly become skilful at using the application with practice.

6.2.3 Admin User Interface

The following themes arose from interviews with expert participants around the archive management extension to Simple DL's admin interface.

6.2.3.1 Theme 1: Ease of Use

As in the general UI, the theme of ease-of-use emerged due to the frequency of statements on it and the simplicity of the system. This theme emerged in part due to the structure of the converted USE questionnaire providing the basis for the interviews. During interviews, all five participants agreed on the management system being easy to use and emphasised the simplicity of the system. One participant noted that "there are not many complexities added, which could distract you from the main purpose which is adding and editing archives." Another stated that the most positive part of the feature was its simplicity, saying that "straightforward and easy to follow". User-friendliness was also positively received with all participants confirming it, and one stating that "the design is clear and simple. It's got an easy user journey." While feedback was positive, two participants did suggest a few ease-of-use improvements. One participant stated an improvement to make usage more obvious through additional feedback, suggesting to "[disable] buttons when an action is done".

6.2.3.2 Theme 2: Ease of Learning

The ease-of-learning theme encompasses all feedback that highlights the ability to learn and remember the admin interface quickly and easily. This theme arose from the predefined grouping of interview questions. All participants' answers spoke positively about the ease and speed of learning involved in adopting the Admin UI. Participants made comments such as "it's easy to remember the steps" and "Memory isn't needed. It's self-explanatory." No negative feedback was given relating to the ease of learning.

6.2.3.3 Theme 3: Archive Description Clutter

Archive-Description-Clutter describes a criticism by two participants that too much information is immediately displayed when users enter the archive managing page. Both participants highlighted concerns around the amount of information displayed for each archive on the manager page, even though all participants agreed on the pleasantness of the overall experience. One participant suggested that some information only be displayed on selection. The participant said, "collapse it so that all versions aren't shown so it doesn't become too cluttered". Additionally, a similar concern was raised by another participant that if some sites had lots of versions, it might make navigation very difficult without hiding some of the information displayed.

6.2.3.4 Theme 4: Stop Scraping Action

The Stop-Scraping-Action theme encompasses an issue two participants found with the suggested method for stopping the scraping of an archive. This theme was mentioned by one participant as an inconsistency of the system due to the removal of archives from the list of displayed archives on the management page after the stop scraping button was clicked. A suggestion was made that it would be better if more instructive feedback was provided after interaction. The participant said that "[you should show] a test notification at the top so you can see what happens." The other participant raised the criticism because it was the one part that would not be easily recoverable in the event of a mistake. The participant said: "[you] can't recover from [stopping] archive scrapes".

6.2.3.5 Theme 5: Multiple Admin Page

The Multiple-Admin-Page theme describes the concerns of two participants around the admin page navigation, as the archive manager is accessed as a link from the default Simple DL admin page. One participant took issue with the choice of name for the archive manager since they believed that it should be the name for the full admin system. Another participant stated that their expectation was that there should only be a single admin page. They said that "the archive management page and file management pages should be together." This feedback highlights potential weaknesses in the usability of the admin UI that could be addressed in future improvements.

6.3 Reflection

The interview format was interesting and informative in how it brought to light vastly different data than if the questions had been distributed as a survey. In this format, interviewees could clarify the terms used in the questions and check which parts of the UI were being referred to in the question. In a survey any confusion would have been unaddressed and could have led to inferior results. One downside of the interview process was the duration that it takes to carry out interviews and the logistics involved in arranging interview slots. However, in future work this could be addressed by having a longer evaluation period with more researchers carrying out interviews simultaneously. One part that could be done differently in future testing would be to use fewer and more in-depth questions, which also directly explain to the participant which part of the UI is being asked about and its location in the application. The reason for this would be to increase the focus on the interviewees feelings and opinions rather than having too many standard questions that may anchor responses based on the questions' wording, which may have occurred in this research. Additionally, more should be done to guarantee that it is explicit what is being examined of the UI for each question, which can be done with descriptions rather than the researcher's chosen names for UI components. For future replication of this work, the recommendation would be to make use of a comparison either by showing an original archive UI or providing two methods of applying the same feature as it might help participants to make relative judgements of their experiences.

6.4 Limitations

A few limitations existed for this research, the biggest of these was around the short time that the research had to be conducted in and the size of the research team. These constraints limited the potential work to only a prototype UI rather than a broader fully integrated archive of archives. Due to the isolated development of the UI in relation to the scraping and ingesting of archives, certain functionality of UI features may behave slightly differently in the proposed prototype to what they would in a fully integrated and completed system. For instance, certain elements of the hierarchy across archive versions and items may change slightly depending on how they are stored and represented in the underlying backend. However, this would not affect the proposed features but rather how or when users interact with it and thus not influence the results of the research.

The UI may also have been limited by being an extension of Simple DL, as this constrained the architecture and limited the interface to being predominantly static. However, at the same time this allowed the research to focus on the features that are required of an archive of archives without re-inventing the wheel and rebuilding standard archiving features. Additionally, Simple DL provided unique features that enabled core parts of what a proposed archive of archive would include, such as support for low resource environments. On top of this, the ongoing experimentation around Simple DL increases the likelihood that this work will see continued research extending it.

7 Conclusions

This research was conducted to propose the first UI for an archive of archives and justify it for future systems through evaluation of its usability. The suggested UI made use of three key features that modified a normal archive to the higher-dimensional equivalent. These features were the archive visualiser, the archive timeline (displaying different versions), and an enhanced admin interface specifically for adding and managing archives. All were implemented as an extension of the Simple DL archiving toolkit. Eleven participants were interviewed for feedback on the usability of non-admin features. Five participants with archiving and admin interface expertise were interviewed for usability feedback on the archive management features. The findings of the interviews were mostly positive, and participants overall found the system very usable, being both easy to use and easy to learn. Additionally, most participants were satisfied by the functionality of the features, and how they met expectations. These results provided strong evidence supporting the underlying research question and promoting the use of the proposed features in the UI of a future archive of archives. The only negative feedback was around the aesthetics of the UI, which was not a priority of the research and does not affect its future viability.

7.1 Answering the Research Question

The research question specifically sought to establish how effectively a user would be able to interact with the interface for an archive of archives. Interview driven usability testing was chosen as the way of evaluating this research question. It was measured by two different groups of users to simulate the difference in people who would use the potential UI in practice. The results on the most part positively answered the research question, particularly around the ease of use and ease of learning criteria used.

7.1.1 Archive Previewing and Archive Timeline UI

The first two proposed features of an archive of archives had two major positive themes that emphasised their usability. Across the usability measurements of ease of use and ease of learning these features received a lot of positive feedback, that indicated that users could effectively interact with it despite it being a higher order archive. Such results contribute to the conclusion that there is a high degree of usability for these features. Despite styling concerns most users were satisfied with the functionality, which further emphasises the high degree of usability of the interface. No glaring usability issues were identified that would signal that users would not be able to interact with the proposed UI or that functional aspects behaved unexpectedly. Thus, the findings considerably met the expectations of the research question, while identifying style as an area that could be a focus of a production system.

7.1.2 Archive Management UI

The archive manager had overwhelmingly positive usability results. Every participant found the system easy to use and easy to learn. These two areas make up the bulk of the usability criteria tested and provide a very strong justification for the research outcomes, which sought to identify if a user could effectively interact with the proposed features. A few areas emerged that highlight potential improvements in the future, such as the way data was displayed, which reduced the satisfaction of some participants. However, since these concerns were not of a functional nature they can easily be modified in later work. Additionally, concerns around the Stop-Scraping-Action and Multiple-Admin-Page themes came about by the current lack of backend integration, which resulted in collections of stored archives being managed with a separate UI to the archive scraping manager for the same archives. While these concerns may impact usability they can be addressed in a better integrated interface. In totality, the most important part of the archive manager, to add archives, received only positive feedback showing that it met the usability requirement set out in the research question, while also providing the basis for further research.

7.2 Problem Area

The general findings were positive and favourably answered the research question. However, the satisfaction element of usability, which relates to the pleasantness of interacting with the design, had a few weaknesses. Participants had concerns around styling and the presentation of certain information. This could have been addressed with more time. However, it does not detract from the core usability of the application specifically with regards to the suggested features as they can always be restyled. The results show that the core functionality is easy and simple to learn and use, which will be more vital for a future archive of archives prototype than finding the perfect style.

7.3 Archive of Archives User Interface Viability

This research has further opened the door for an archive of archives because the usability of prototype was well received. All three main features were achieved positive usability feedback from participants and provided novel functionality for an archive of archives. The strong usability results indicate that the intended outcome of creating a UI that could be effectively interacted with was accomplished. Although participants had mentioned styling concerns these had minimal effect on usability. Overall, the results are conclusive that the archive of archiving interface proposed is usable and can be taken forward, even if room for improvement exists.

7.4 Relevance of Results

These results are important in providing the basis for future research into the novel and experimental concept of an archive of archives. This contributes to the forefront of digital library research and helps create a prototype for a completely new computing interface that aids the movement to find a solution to the disappearance of information.

7.5 Future Work

Overall, this work provided a suitable prototype for how a future archive of archives UI could be built around the key features of visualisation, versioning, and administration. However, this work can be taken forward though implementation, integration, further extensions, and more research on the overall archive of archives concept.

7.5.1 Archive of Archives Integration

Future work could be performed to create a fully operational archive of archives with an integrated backend for the UI. This could examine potential issues that might arise or affect features when put into production. Additionally, the system could be tested as a complete prototype beyond only the usability of its UI. Completing such research is crucial to eventually putting an archive of archives into practice. The current built model could be integrated into a Simple DL archive of archives with other research and development around the scraping and ingestion for an archive of archives.

7.5.2 Aesthetics and Design Focus

This work focused on function over aesthetics; future work could attempt different ways of visualising these features and spend greater time on the aesthetics side. Additional aesthetic improvements could be made to the functional UI features implemented in this research. Potential UIs can be created and experimentally compared by participants to find the best design for an Arch2 UI.

7.5.3 Additional User Interface Features

Future work could be done to propose additional features that a particular archive of archives would need in its user interface. Already, recommendations were made by research participants suggesting certain ideas such as the need for programmatic back buttons to help navigate the website previews/views of archives as the browser's one is inconvenient to use. On the admin side it was also recommended by a participant to add manual scraping capabilities, such as the pause and resuming of scraping, since some archives might not need to be scraped as frequently or may not want the regular high traffic of being scraped. Additionally, advanced scraping configuration could be added to the UI.

References

[1] Carbajal, I. A. and Caswell, M. Critical Digital Archives: A review from archival studies. *The American Historical Review*, 126, 3 (2021), 1102-1120.

[2] Yadav, D. Opportunities and challenges in creating digital archive and preservation: an overview. *International Journal of Digital Library Services*, 6, 2 (2016), 63-73.

[3] Hedstrom, M. Digital preservation: a time bomb for digital libraries. *Computers and the Humanities*, 31, 3 (1997), 189-202.

[4] Pandey, R. *Digital library architecture*. 2003.

[5] Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R. and Walker, J. H. DSpace: An open source dynamic digital repository (2003).

[6] Owens, T. *The theory and craft of digital preservation*. Johns Hopkins University Press, 2018.

[7] Jantz, R. and Giarlo, M. J. Digital preservation: Architecture and technology for trusted digital repositories (2005).

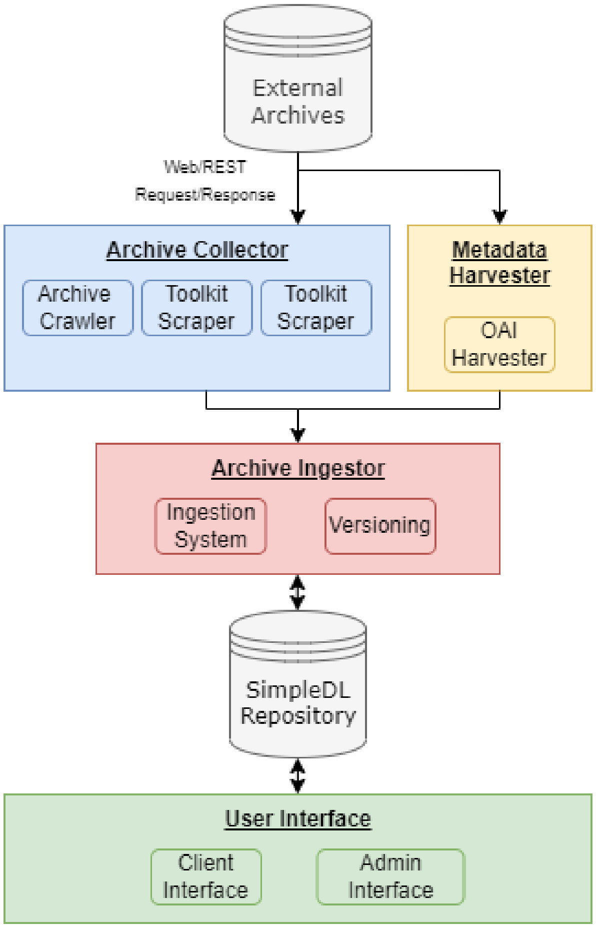
[8] Niu, J. An overview of web archiving. *D-Lib magazine*, 18, 3/4 (2012).

[9] Suleman, H. *Investigating the effectiveness of client-side search/browse without a network connection*. Springer, 2019.
[10] Suleman, H. *Digital libraries without databases: The bleek and lloyd collection*. Springer, 2007.
[11] Xu, W., Esteva, M., Beck, D. and Hsieh, Y.-H. *A portable strategy for preserving web applications functionality*. IEEE, 2017.
[12] Alam, S., Kelly, M., Weigle, M. C. and Nelson, M. L. *Client-side reconstruction of composite mementos using serviceworker*. IEEE, 2017.
[13] Suleman, H. *In-browser digital library services*. Springer, 2007.
[14] Suleman, H. *Simple DL: A toolkit to create simple digital libraries*. University of Cape Town, South Africa, 2021.
[15] Lagoze, C., Payette, S., Shin, E. and Wilper, C. Fedora: an architecture for complex objects and their relationships. *International Journal on Digital Libraries*, 6, 2 (2006), 124-138.
[16] Mayo, C., Jazairi, A., Walker, P. and Gaudreau, L. BC digitized collections: towards a microservices-based solution to an intractable repository problem. *Code4Lib Journal*, 44 (2019).
[17] Amaral, M., Polo, J., Carrera, D., Mohomed, I., Unuvar, M. and Steinder, M. *Performance evaluation of microservices architectures using containers*. IEEE, 2015.
[18] Suleman, H. Reflections on design principles for a digital repository in a low resource environment (2019).
[19] Suleman, H. *An african perspective on digital preservation*. World Scientific, 2011.
[20] Blischak, J., Davenport, E. and Wilson, G. *A Quick Introduction to Version Control with Git and GitHub*. PLoS Computational Biology. DOI:10.1371/journal.pcbi.1004668, 2016.
[21] Jiang, J., Lo, D., He, J. and Xia, X. *Why and how developers fork what from whom in GitHub*. Chinese distribution service of audiovisual products in international copy right trade. DOI:10.1007/s10664-016-9436-6, 2017.
[22] Yu, J. *Research process on software development model*. IOP Publishing, 2018.
[23] Gao, M., Kortum, P. and Oswald, F. *Psychometric evaluation of the use (usefulness, satisfaction, and ease of use) questionnaire for reliability and validity*. SAGE Publications Sage CA: Los Angeles, CA, 2018.
[24] Lund, A. M. Measuring usability with the use questionnaire12. *Usability interface*, 8, 2 (2001), 3-6.
[25] Noble, H. and Heale, R. *Triangulation in research, with examples*. Royal College of Nursing, 2019.
[26] Gill, S. L. Qualitative sampling methods. *Journal of Human Lactation*, 36, 4 (2020), 579-581.
[27] Etikan, I. and Bala, K. Sampling and sampling methods. *Biometrics & Biostatistics International Journal*, 5, 6 (2017), 00149.
[28] Turner, C. W., Lewis, J. R. and Nielsen, J. Determining usability test sample size. *International encyclopedia of ergonomics and human factors*, 3, 2 (2006), 3084-3088.
[29] Hwang, W. and Salvendy, G. Number of people required for usability evaluation: the 10±2 rule. *Communications of the ACM*, 53, 5 (2010), 130-133.
[30] Maguire, M. and Delahunt, B. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *All Ireland Journal of Higher Education*, 9, 3 (2017).
[31] Kiger, M. E. and Varpio, L. Thematic analysis of qualitative data: AMEE Guide No. 131. *Medical teacher*, 42, 8 (2020), 846-854.
[32] Nowell, L. S., Norris, J. M., White, D. E. and Moules, N. J. Thematic analysis: Striving to meet the trustworthiness criteria. *International journal of qualitative methods*, 16, 1 (2017), 1609406917733847.

Appendix

A.1 Archive of Archives Architecture

Below is the architecture proposed for a full archive of archives built as an extension to the Simple DL toolkit.


External Archives
Web/REST Request/Response
Archive Collector
Archive Crawler
Toolkit Scraper
Toolkit Scraper
Metadata Harvester
OAI Harvester
Archive Ingestor
Ingestion System
Versioning
SimpleDL Repository
User Interface
Client Interface
Admin Interface