Marco Filipe
Nunes Soares
Abrantes Pereira

Repositório Digital Pessoal Semântico Baseado na "Cloud"

A Semantically Enhanced Cloud Based Personal Digital Repository

Marco Filipe
Nunes Soares
Abrantes Pereira

Repositório Digital Pessoal Semântico Baseado na "Cloud"

A Semantically Enhanced Cloud Based Personal Digital Repository

Tese apresentada às Universidades de Aveiro, Minho e Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Informática, realizada sob a orientação científica do Doutor Joaquim Arnaldo Martins, Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Trabalho financiado pelas seguintes entidades:
Este trabalho foi financiado em parte pela bolsa SFRH/BD/62554/2009 da Fundação para a Ciência e a Tecnologia

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA
COMPETE
POPH QUALIFICAR É CRESCER.
QREN QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
GOVERNO DA REPÚBLICA PORTUGUESA
UNIÃO EUROPEIA Fundo Social Europeu

o júri / the jury

presidente / president

Doutor Luís Castro
Professor Catedrático da Universidade de Aveiro (por delegação do Reitor da Universidade de Aveiro)

vogais / examiners committee

Doutor José Luís Brinquete Borbinha
Professor Associado, Instituto Superior Técnico, Universidade de Lisboa

Doutor José Eduardo de Mendonça Tomás Barateiro
Investigador Auxiliar, Laboratório Nacional de Engenharia Civil, Professor Auxiliar Convidado, Universidade Nova de Lisboa, Nova IMS

Doutor Gabriel de Sousa Torcato David
Professor Associado, Faculdade de Engenharia, Universidade do Porto

Doutor José Manuel Matos Moreira
Professor Auxiliar, Universidade de Aveiro

Doutor Joaquim Arnaldo Martins (Orientador)
Professor Catedrático, Universidade de Aveiro

agradecimentos

Em primeiro lugar gostaria de agradecer ao meu orientador, o Professor Joaquim Arnaldo Martins pela oportunidade, orientação e apoio concedido durante o meu doutoramento. Este trabalho só foi concluído graças a todas as suas intervenções que me colocaram sempre no caminho correto. Gostaria também de agradecer aos amigos, principalmente ao João Pereira, Luís Ribeiro, Luís Matos e Pedro Goucha, que de uma forma ou de outra me foram encorajando ou ajudando quando foi necessário. Por último, mas não menos importante, agradeço à minha família, que me apoiou de forma incondicional desde o inicio desta odisseia.

acknowledgements

First, I would like to thank my supervisor, Professor Joaquim Arnaldo Martins for the oportunity, guidance and support given throughout the Ph.D. This work could only be done due to all of his interventions that always placed me in the right path. I would also to the friends, particularly to João Pereira, Luís Ribeiro, Luís Matos e Pedro Goucha, that in one way or another encouraged or helped when it was needed. Last, but not least, I thank my family that has always and unconditionally supported me from the beginning of this odyssey.

Palavras-chave Repositório Digital Pessoal, Recolha de Conteúdo, Semântica

Resumo

Ao longo do tempo os indivíduos procuraram sempre formas de preservar o conhecimento, recordações e experiencias de vida. A busca por suportes estáveis que possam preservar as recordações dos efeitos da passagem do tempo leva à projeção das mesmas sobre objetos físicos. Estes objetos eventualmente são agregados em coleções que representam partes das vidas dos seus criadores, e que que podem ser partilhadas com outras pessoas. O uso generalizado das tecnologias da informação, conjuntamente com a sua simplicidade trouxe consigo uma mudança de paradigma, levando a que muitas interações que poderiam criar objetos físicos sobre os quais seriam projetadas recordações passassem do mundo físico para o mundo digital passando a criar objetos digitais, sobre os quais também podem ser projetadas recordações, tal como o que acontece com os seus equivalentes físicos. Devido a sua natureza digital estes objetos são simples de criar, manipular, duplicar e partilhar. Estas características colocam-nos numa posição em que podem ser gerados facilmente, usado para transmitir conteúdo aparentemente trivial que é depois partilhado e prontamente esquecido. No entanto, apesar destes objetos poderem passar a incorporar memorias, a combinação do excesso de confiança nas suas características intrínsecas e de uma atitude que convida ao esquecimento acabam por impedir este desfecho, o que pode levar a que no futuro os indivíduos percam o acesso a estes objetos. O trabalho desenvolvido ao longo desta tese foca-se sobre este problema, propondo resolve-lo com a criação de um sistema de repositórios digitais pessoas para a recolha de informação sobre o conteúdo pessoal de cada individuo. Em vez de se focar na recolha do conteúdo propriamente dita, um repositório digital pessoal dá prioridade à recolha de metadados sobre o conteúdo (desde que este não esteja em perigo iminente) de forma a no futuro poder guiar os indivíduos de volta aos serviços na "nuvem" onde o conteúdo ainda reside no seu contexto original. Em cenários pessoais não é viável recorrer a pessoal especializado para proceder a recolha e seleção destes dados. Para mitigar este problema, os dados são recolhidos o mais cedo e próximo da origem quanto possível por agentes de recolha. Estes foram desenhados de forma a minimizar a intrusão nas rotinas dos seus utilizadores, ao mesmo tempo que oferecem serviços complementares que podem ser utilizados de forma independente do repositório digital pessoal, fomentando assim a adoção do uso destes agentes. Este trabalho também descreve uma proposta de extensão ao modelo CIDOC/CRM, utilizado para classificar e organizar a informação recolhida. Esta extensão foi criada devido à necessidade de dotar o modelo de novas entidades e propriedades destinadas a lidar com objetos digitais e cenários pessoais.

Keywords Personal Digital Repository, Content Gathering, Semantic

Abstract

Throughout time individuals have always sought forms to preserve their knowledge, memories and life experiences. Physical objects provide a medium upon which individuals are able to project their memories, in an attempt that they remain in a stable support better able to cope with the passage of time. Physical objects eventually coalesce into a collection that comes to represent part of its owners' lives and that can eventually be passed on to others. Widespread use of information technologies, coupled with their perceived ease of use has shifted many interactions that would end up producing external memory objects from the physical to the digital realm. As with their physical counterparts, digital objects can also be used by individuals to project their memories. Due to their digital nature, these objects are simple to create, produce, manipulate, duplicate and share. These traits place them into a position where they can be generated without too much effort to convey what might appear to be trivial content, readily shared and forgotten afterwards. Though, through memory projection they could become part of their creator's legacy, overconfidence in their reproducibility and being forgotten can prevent them from being so. This deprives their creators from part of their lives that, in spite of appearing trivial at first, might acquire a deeper meaning with the passage of time. The work done throughout this thesis addresses this issue by proposing the creation of personal digital repositories to collect information regarding personal content. Instead of focusing on collecting the content itself, the personal digital repository prioritises gathering metadata about the content (when not immediately at risk) in order to lead its owner back to the "cloud" applications where the content can still be found in its original context. In personal scenarios it is not feasible to rely on trained personnel to help with content gathering and organisation. To mitigate this issue, content is collected as soon as possible by collection agents. These are designed to be as unobtrusive as possible, also offering additional services that can be used even without the personal digital repository in order to encourage their adoption. This creates an intertwined ecosystem where the content collection agents feed the personal digital repository and can in turn use previously collected content to support their additional services. This work also describes a proposed extension to the CIDOC/CRM model, used to classify and organise the collected information. The extension was created due to a perceived gap in the CIDOC/CRM model when it came to dealing with digital objects.

Contents

List of Figures

List of Tables

Acronyms

5S	Streams, Structures, Spaces, Scenarios and Societies
AIP	Archive Information Packages
API	Application Program Interface
CIDOC/CRM	CIDOC Conceptual Reference Model
CSS	Cascading Style Sheets
DC	Dublin Core
DIP	Dissemination Information Packages
DIY	Do It Yourself
DOI	Digital Object Identifier
DRM	Digital Rights Management
DVR	Digital Video Recorder
EDM	Europeana Data Model
EPDR	Entity - Personal Digital Repository
ER	Entity-Relationship
ESE	Europeana Semantic Elements
FOAF	Friend Of A Friend
FRAD	Functional Requirements for Authority Data
FRBR	Functional Requirements for Bibliographic Records
$FRBR_{oo}$	FRBR - object oriented
FRSAD	Functional Requirements for Subject Authority Data
GIF	Graphics Interchange Format
GPS	Global Positioning System
GSM	Global System for Mobile Communications

HOPPLA . . . . . Home and Personal Persistent Long term Archiving

HTML . . . . . HyperText Markup Language

HTTP . . . . . HyperText Transfer Protocol

IRC . . . . . Internet Relay Chat

ISBD . . . . . International Standard Bibliographic Description

ISO . . . . . International Organization for Standardization

ISSN . . . . . International Standard Serial Number

JSON . . . . . JavaScript Object Notation

LOCKSS . . . . . Lots of Copies Keep Stuff Safe

MIME . . . . . Multipurpose Internet Mail Extensions

MMS . . . . . Multimedia Messaging Service

OAI-PMH . . . . . Open Archives Initiative Protocol for Metadata Harvesting

OAIS . . . . . Open Archival Information System

OCR . . . . . Optical Character Recognition

ORE . . . . . Object Re-use and Exchange

OWL . . . . . Web Ontology Language

PDF . . . . . Portable Document Format

PIM . . . . . Personal Information Manager

PPDR . . . . . Property - Personal Digital Repository

RDF . . . . . Resource Description Framework

RDFa . . . . . Resource Description Framework in Attributes

RDFS . . . . . Resource Description Framework Schema

RELS-INT . . . . . Relations Internal

RELS-EXT . . . . . Relations External

REST . . . . . Representational State Transfer

SaaS . . . . . Software as a Service

SAIL . . . . . Storage and Inference Layer

SIP . . . . . Submission Information Packages

SMS . . . . . Short Message Service

SPARQL . . . . . SPARQL Protocol and RDF Query Language

SSH . . . . . Secure Shell Protocol

URL . . . . . Uniform Resource Locator

VoIP . . . . . Voice over IP

XML . . . . . Extensible Markup Language

Chapter 1

Introduction

"There is no man (...) however wise, who has not at some period of his youth said things or lived a life, the memory of which is so unpleasant to him that he would gladly expunge it. And yet he ought not entirely regret it, because he cannot be certain that he has indeed become a wise man - so far as it is possible for any of us to be wise - unless he has passed through all the fatuous or unwholesome incarnations by which that ultimate stage must be preceded... We do not receive wisdom, we must discover it for ourselves after a journey through the wilderness which no one else can make for us, which no one can spare us, for our wisdom is the point of view from which we come at last to regard the world"

Marcel Proust

Remembrance of Things Past: Volume I - Swann's Way & Within a Budding Grove

Throughout history mankind has sought forms of passing knowledge from generation to generation. As individuals we attempt to preserve our memories and leave a mark upon the world in order to avoid be forgotten. From painting in caves to the invention of writing; from painted portraits to photography, the evolution of technology has provided us with novel supports to create what psychologists describe as "*external memory*" [1, 2]. The objects that compose this *external memory* are the physical embodiments of our memories and knowledge. They ensure that our memories remain in a (somewhat) stable form that is better able to cope with the passage of time without compromising the memories they hold (unlike our own memory, that with the passing of time tend to slightly change our perception of past events). At the same time, relying on these physical embodiments actually encourages us to remember less things at any given time [2], since we can use them to support our own memory. Albert Einstein provides anecdotal evidence of this behaviour, as he is credited to have said "*I do not carry such information in my mind since it is readily available in books*", when asked what was the speed of sound [3]. *External memory* objects eventually coalesce into collections that represent bits and pieces of our individual lives. While some of these collections become relevant enough (due to its size, or by being amassed by historical figures) to be gathered by memory institutions such as museums, archives or libraries, most of them remain part of our personal heritages, with some eventually becoming family heirlooms.

Technology is not only responsible for providing novel supports that we can use as *external memory* holders, but also for the democratisation of their use. As the *external memory*

suppports become more affordable and easier to use, our ability to amass a collection of objects also increases. Not so long ago, only the wealthy could afford to commission their portraits in order to be seen through the ages, yet the invention of photography has brought this privilege to everyone. Likewise the rise in literacy has taken what was once the privilege of an elite, the creation of written texts, and transformed it into a universal right that is able to to fill countless *external memory* objects, ranging from personal letters to literary masterpieces. *External memory* objects remained physical embodiments of our knowledge until the advent of mass access to both personal computing devices (regardless of actual form factor) and the Internet. The combined effect of those two technologies transformed the way we live and more importantly the way *external memory* objects are produced. While they still embody our knowledge, many of them exist in pure digital form, that we create and access through "*gateway*" devices. This implies that this class of *external memory* objects are no longer associated with a specific physical object (tough this does not mean that they don't need a physical support) since they can be accessed and manipulated in myriad of different devices. From digital photography to digital documents, the number of "*born digital*" objects is increasing. Where once our objects were kept just for ourselves or our family, with access to affordable devices that can create (and consume) content from nearly everywhere and the Internet as a relatively low-cost storage and distribution channel we can make our personally created objects available to global audiences and in the same way access to content created by others nearly instantaneously. A report [4] indicated that Americans consumed around 34 gigabytes of information each year (excluding information consumed while at work) delivered by different means (TV, radio, printed press, computers), while a different study places the amount of information communicated in 2007 by mankind as whole at 67 exabytes, or the equivalent of 6 newspapers per person on the planet [5]. This unprecedented access to content creates an information cycle, where we use content gathered from various mediums (TV, radio, printed press, personal digital devices) to form opinions, which we them externalise to digital *external memory* objects that eventually we share with others that in turn will use them to generate more digital *external memory* objects thus perpetuating the cycle.

As more and more activities from our daily lives require interaction with information technologies, the importance of digital objects as *external memory* holders increase. Unlike their physical counterparts, we can create multiple copies of a digital object with a (relatively) minimal investment in both time and effort. New copies will be indistinguishable replicas of the original, a trait that indirectly promotes true sharing over lending since it removes the penalty (i.e. being deprived of the object for the duration of the lend) imposed to the lender in favour of the creation of a new copy. At the same time this trait can create a false sense of security. Being easily created, replicated and without a sharing penalty, digital objects can lead digital objects to be seen as trivial or even disposable. Some digital objects are created with the sole purpose of serving as a sharing vehicle, or communicating information that at the time does not appear to be relevant enough to ever create a true memory. As a result of their intended use, these digital objects eventually become lost within the Internet or in personal devices. Though some might say that the Internet never forgets [6], we as individuals most surely do. With so many digital objects being created and shared it becomes increasingly difficult to keep track of where we have placed them. This is something that digital objects have in common with their physical counterparts: they can be misplaced or downright lost (no matter how many copies one creates), leading to loss of information. Even when we know exactly where they are it is still possible to lose the ability to use them, because the software or the hardware required to use them is no longer available. Vinton Cerf illustrates this issue with a simple

example: he uses Microsoft Office 2011 on a Mac, but the software is unable to interpret a PowerPoint file created with the 1997 version of Office since "It doesn't know what it is" [7]. This is a well known research problem, that has been given the name of "*digital dark age*" [8]. While some are worried that in one way or another we can forfeit our digital heritage, others have the opposite concern. Based on the assertion that the Internet never forgets, Eric Schmidt has expressed the concern that the "*Internet generation*" might not be allowed to forget the "mistakes" they made and shared online when they were teenagers [9]. Motivated by similar worries, particularly about the privacy of its citizens, the European Union has a proposal to regulate the collection and retention of personal data, which in article 17 enshrines the "*right to be forgotten*" as part of its General Data Protection Regulation [10]. These contradictory concerns show us that we stand at a peculiar crossroads: we are simultaneously worried that the Internet might never forget anything and that it won't remember everything.

On first glance, we appear to need to choose between a "*big brother*" style Internet that never forgets and a "*goldfish*" Internet that is only able to recall selected, *sanitised* parts of our digital lives. Though appearing to be only choices available at the moment, they end up representing an oversimplification of the issue at hand. The "*right to be forgotten*" is interesting on theory, but its actual implementation might be problematic. After all, even if an individual addresses a request to the service (or services) to where he managed to upload his "mistake", how can those in charge ensure that every single copy will be stricken from the records, especially if we include in those records any backups made as part of the daily operations of said services? It is not yet feasible to control every copy made by other individuals (that will proceed to share it using other services) while the "mistake" was publicly accessible. Placing the burden of scouring the Internet in search of every last copy of something deemed as a "mistake" in a single individual (even if that individual is the interested party) more than not being an optimal solution, it appears to be some sort of punishment for having committed and publicised that "mistake". On the other hand, if the Internet never forgets but individuals do, there is always the risk of being caught off guard when someone stumbles upon their "mistakes". Yet the same things that have been labelled by Eric Schmidt as "mistakes" are undeniably part of our personal memories (not only those "mistakes" but every digital *external memory* objects). They helped to shape who we are and show the strength of our convictions (no matter how misguided they appear to be from a vantage point in the future). By creating and disseminating digital *external memory* objects we are collectively (albeit unwittingly) using the digital space as a mixture of personal showcase and warehouse for our memories. Taking all of this into account, the choice at hand might not be how forgetful we want (or need) the Internet (and our personal devices) to become, but instead how can we use the warehouse we have been given as an effective personal repository for our digital *external memory* objects, while dealing with its implicit showcase nature.

The research conducted during the course of the doctoral programme is aimed at the creation of an environment (i.e. a repository) that can be used by individuals to maintain their digital *external memory*, represented by the various digital objects created through the course of their daily lives. Unlike traditional repositories, that attempt to acquire and store the digital objects in a centralised location, branding the acquired objects as the authoritative versions, this work adopts a different approach that allows digital objects to remain in their natural context whenever possible. Thus instead of a repository comprised primarily of digital objects themselves, this work proposes the creation of a meta repository, whose information can be used to lead individuals back to their digital objects (which are assumed to be scattered through multiple services) should the need ever arises. This approach is particularly suitable to

deal with digital objects that are either purposely shared through online services or that exist within closed platforms (also known as "walled gardens"), since it does not require absolute control over the digital object (allowing it to continue to evolve on its own) and encourages individuals to continue to use their preferred services to store and share their digital objects. Nevertheless the goals of this research can be summarised as follows:

- Develop a strategy for non-intrusive content collection. Digital objects in personal scenarios tend to be scattered throughout multiple platforms and devices. Traditional content collection strategies tend to rely either on dedicated personnel for content gathering and selection or in voluntary content submissions, both of which are ineffective strategies in personal scenarios. Regardless if it stores the digital objects themselves or pointers and metadata to lead its owner back to said digital objects, the success of a personal digital repository hinges on the adoption of novel content collection strategies that do not alienate its intended user base by being too intrusive or complex.

- Develop a flexible model to support the personal digital repository. The role of a traditional digital repository is to support a given community, that shares a specific interest. To accomplish this task, traditional repositories are designed to cater the needs of that specific community, and consequently designed around the shared interest. While a personal digital repository will still serve the needs of a particular community (though in this case the community is composed of a single individual who owns the repository) it will also need to deal with that individual's multiple interests, which means it cannot be designed with a specific topic in mind. Furthermore, both the areas of interest as well as the level of engagement are more likely to change over time in personal scenarios, in contrast with what happens in the scenarios where traditional digital repositories are deployed, where is more likely for the members of the community to change over time.

- Collect non conventional digital objects which are not covered by traditional repositories. Traditional digital repositories are designed to deal with content heavy digital objects, like documents, photos or videos, offering little to no options to deal with other digital object types such as web browsing history, social network graph and posts or even the ubiquitous email. These object types are sometimes seen as a by-product of the information age (particularly social network interactions). Individually, a single digital object of one of these types might not be particularly interesting, or even relevant, yet as a whole they are indicative of its creator's thoughts, interests or work process. If we take into account that according to multiple market studies [11, 12], individuals spend up to 30% of their online time using social networks and email, it is likely that some of digital objects generated from using those communication tools will contain some piece of cultural or personal information that can at the very least be used to further contextualise a more traditional digital object.

1.1 Organisation

This thesis is divided in six chapters, which besides the current introductory chapter are organised in the following way:

Chapter 2 provides an overview of the existing approaches, methodologies and standards related to the field of digital repositories. It proceeds by presenting existing surveys of

user behaviour regarding digital preservation strategies, which will act as a bridge to establish why the traditional approach to managing a digital repository is unsuitable for use in personal scenarios.

Chapter 3 introduces the proposed model and architecture for a personal digital repository. It provides high level descriptions of the adopted model, required services, extension capabilities and content gathering strategies, as well as how they can contribute to mitigate or eliminate some of the issues associated with the usage of digital repositories in personal scenarios identified in the previous chapter.

Chapter 4 describes the ontologies used to support and organise the personal digital repository. This chapter approaches both extensions made to existing ontologies to support the personal digital repository as well as the interactions and mappings established between those used in the personal digital repository and other, existing ontologies that are not directly used.

Chapter 5 portraits a use case for a personal digital repository, in which the use of the personal digital repository and its surrounding content gathering agents allow an individual to generate a description of their daily digital activities and produced (or "acquired") digital objects.

Chapter 6 is dedicated to the discussion of the personal digital repository as a whole. It also presents some pointers about how it can be improved in the future.

Chapter 2

State of the art

As information began to appear in digital supports, researchers had once again to answer the question of how to manage a growing corpus of information on a new arena. Some of the techniques used by physical archives and libraries could be adapted and reused, particularly during the digital age's early days when managing digital information meant primarily managing its physical supports. As technology continued to evolve it opened up new possibilities for organising, managing and ensuring access to digital objects, while at the same time it exposed new challenges and incompatibilities between technological generations that required novel solutions. On one hand digital repositories, archives and libraries became accessible around the clock, gave their users the possibility to search (while expecting results in a reasonable amount of time, that often means instantly) not only in summaries but within the actual content of digital objects and by creating links between related pieces of information within digital objects (that in some cases may even be to external sources) streamlined the navigation in said information. On the other hand, digital objects still need to be collected, treated and organised in order to became useful within these systems, yet subtle (and sometimes not so subtle) differences in the organisation schema or in its interpretation and the use of distinct versions of the same file format (or even different formats) for low level representation still hinder the exchange of information (and with it the establishment of meaningful links) between systems and in the long run difficult the task of preserving digital objects for future use. Some of these challenges are addressed in a number of models, with varying scopes, that can be adopted and used in the tools used to create these kind of systems. This chapter is dedicated to providing an overview of the available tools, models, and notable research efforts being done regarding these areas of digital information management.

2.1 Models

There are different types of models that can be applied to digital repositories. Some describe the functionalities, responsibilities, and structure of the repository while others describe the organisation of the digital objects stored in the digital repository. Given the multitude of subjects that the objects contained in a repository might cover, high-level reference models for digital repositories, whose main concern is to provide a global description of what a digital repository should be, usually avoid delving too deep into how the digital objects stored in them will be organised, or what kind of specific metadata should be provided, and as a result they can only be used as guidelines when assembling a repository. Mid-level conceptual

models provide frameworks over which digital objects can be overlaid, thus establishing the repository's data structure, internal organisation and classification schemes. These type of models can sometimes be applied directly, though they still require slight adjustments and fine tuning in order to support the functionality defined (or required) by the reference model and/or extensions in order to better describer specific types of digital objects. This section provides an overview of three high level reference models Open Archival Information System (subsection 2.1.1), DELOS Digital Library Reference Model (subsection 2.1.2) Streams, Structures, Spaces, Scenarios and Societies (subsection 2.1.3) and four conceptual models Functional Requirements for Bibliographic Records (subsection 2.1.4), ABC Ontology and Model (subsection 2.1.5), Europeana Data Model (subsection 2.1.6) and CIDOC Conceptual Reference Model (subsection 2.1.7). The reviewed reference models present different perspectives on the goals and missions that developers should take into account when developing a new repository or digital library, while the conceptual models present different strategies that can be applied to classify and organise digital objects within a repository. The terms digital repository and digital library are often used interchangeably, yet if one thinks carefully about them it is possible to find a subtle but critical distinction. While both can be seen as gathering points for digital objects, thinking about libraries can also evoke the figure of the librarian, and part of the librarian's task is to select what is accepted onto the library, creating a very controlled environment. On the other hand a repository repository.does not possess that association and is more often associated with free contributions. While this work is focused on personal digital repositories, whose content acceptance policy might on limit be described as "*anything the repository owner wants*", the views and needs required to support librarians can still provide worthy insight to support the design of a personal digital repository.

2.1.1 Open Archival Information System

Despite the name, a digital repository is more than just a set of digital objects and the systems designed to store, preserve and ensure access to those digital objects. As with their physical counterparts, when thinking about a digital repository one should also consider the communities that surround it and are crucial for its function. From the producers of digital objects, to the management team, to end users every member of these communities has a stake in the digital repository and as such they became a part of it. The Open Archival Information System (OAIS) [13], which has also been defined as an International Organization for Standardization (ISO) standard (ISO 14721:2003), takes into account these multiple components in order to provide a platform agnostic reference model for repositories and archives that rely primarily on digital information. It provides a set of models, conceptual definitions and processes that can be used as guide when creating new repositories, or assessing existing ones. In OAIS, digital repositories have a set of responsibilities that they are required to fulfil in order to be considered compliant with the model:

- Negotiate for and accept appropriate information from information producers.
- Obtain sufficient control of the information provided to the level needed to ensure Long Term Preservation.
- Determine, either by itself or in conjunction with other parties, which communities should become the Designated Community and, therefore, should be able to understand the information provided, thereby defining its Knowledge Base.

- Ensure that the information to be preserved is Independently Understandable to the Designated Community. In particular, the Designated Community should be able to understand the information without needing special resources such as the assistance of the experts who produced the information.

- Follow documented policies and procedures which ensure that the information is preserved against all reasonable contingencies, including the demise of the Archive, ensuring that it is never deleted unless allowed as part of an approved strategy. There should be no ad-hoc deletions.

- Make the preserved information available to the Designated Community and enable the information to be disseminated as copies of, or as traceable to, the original submitted Data Objects with evidence supporting its Authenticity.

As with a traditional repository, a digital repository stores more than just the raw digital object. The additional meta data required to ensure provenance, authenticity, and context among other things must also be stored by the digital repository. OAIS provides an information model that can be used as a reference when implementing new repositories. At the core of the OAIS information model is the information object, which contains both the digital object and the representation information. From this top level object, others might be derived according to their intended function within the repository. Within an information object, the representation information is the information required to correctly interpret a digital object from its underlying bit stream, and is composed of structure information (also known as the digital object's format), semantic information which attributes meaning to the information decoded from the digital object's bit stream using the structure information and additional representation information (such as any specific software required to decode and interpret the digital object). Since representation information is itself stored in a digital format, it can be itself considered to be a digital object, and as such be required to have representational information, thus creating an representation network. It should be noted that the complete description of the representation object does not have to reside in the OAIS compliant repository, and might be stored at an external trusted repository. In this case the representation information will contain a pointer to this external repository. Regardless of its type, the OAIS model makes suggestions on how what types of information objects should be packaged together for different tasks. It suggest the use of Submission Information Packages (SIP) for the submission process, Archive Information Packages (AIP) for long term storage and Dissemination Information Packages (DIP) for content retrieval by the repository users.

In order to fulfil the previously mentioned responsibilities, the OAIS model suggests a set of functionalities (seen in Figure 2.1) that should be implemented by a digital repository. It should be noted that although the OAIS model splits these functionalities into six distinct (but interacting) groups, actual implementations might choose to split them in a different way to cope with technological or organisational constraints. Furthermore some of these services don't have to be fully digital in nature and can be provided by the community that manages the digital repository.

Ingestion the set of services and functionalities required to accept submission packages into the repository and prepare them for storage.

Archival Storage the set of services and functionalities required to properly store, maintain

and retrieve archival packages. These functionalities are also responsible for handling with potential unexpected disasters that might threaten the integrity of the repository.

Data Management the set of services and functionalities responsible for maintaining the repository conceptual model, and create, maintain and ensure access to archival package's descriptive information that allows end users to find and retrieve digital objects form the repository.

Administration the set of services and functionalities required to maintain the overall operation of the digital repository. These include high level policy making, formal contracts with the community that submits digital objects to the repository and conducting audits to ensure the quality of the repository's content.

Preservation Planning the set of services and functionalities required to monitor the digital repository, issue recommendations regarding the current state of the repository and advise on migration policies. These functionalities are also responsible to ensure that a stored digital object remains intelligible even if its original computing platform has become obsolete, and for establishing the format of all the information packages that the repository will handle.

Access the set of services functionalities that enable users to interact with the repository in order to retrieve one or more digital objects (encased in dissemination packages), their description or know the object's current state. These functionalities are also responsible for enforcing any type of access control that might be in place, be it repository-wide or applied only to a single digital object.

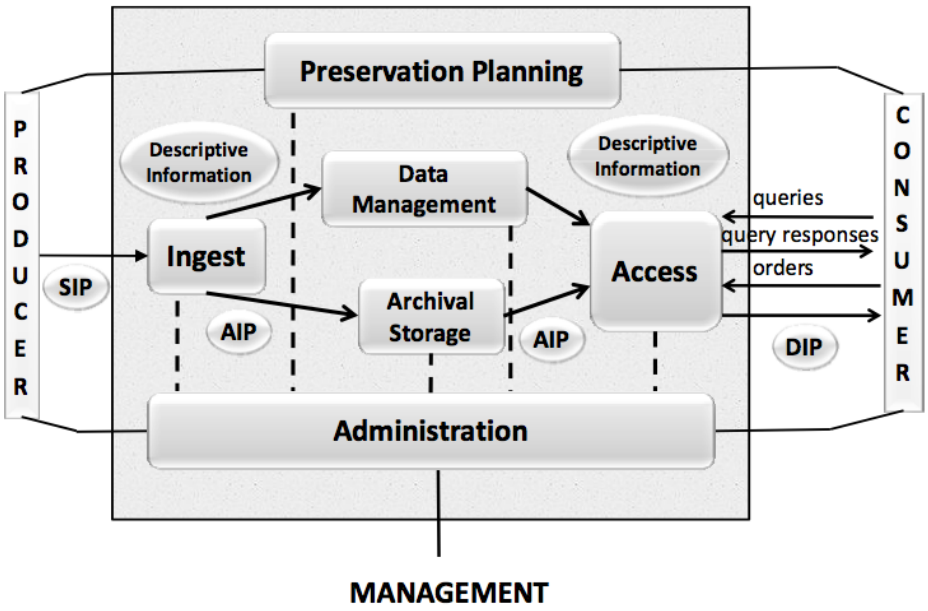


Figure 2.1: OAIS Functional Entities [13]

2.1.2 DELOS Digital Library Reference Model

In the field of digital libraries the DELOS digital library reference model [14] provides a high level view of the multiple components required to create a digital library. This reference model establishes the context required to represent the different facets of digital libraries without prescribing any type of concrete implementation, making it platform agnostic. The model's basic premise is that digital libraries constitute a complex universe, that can be seen from different perspectives or domains. Each domain can itself be seen from multiple levels, from a high level completely conceptual level to the implementation level. The DELOS digital library reference model attempts to integrate all these levels within a single coherent representation using the concept of frameworks. Thus a very high level view of a digital library will encompass multiple frameworks that rely on each other in order to produce a coherent system. From the more abstract to the more concrete, it defines as frameworks the *Reference Model* whose task is to represent the core concepts (used in a particular digital library) relations and axioms inherent to a given problem domain without prescribing any implementation or technology; the *Reference Architecture* whose task is to model in abstract the solution required to implement the concepts present in the reference model and the *Concrete Architecture*, whose task is to take the abstract solution modelled in the *Reference Architecture* and replace them by actual technologies, standards and services that will implement the digital library. This tiered architecture model provides a common ground over which digital library systems (both existing and newly implemented) can be compared. Additionally, within the DELOS digital library reference model a digital library is not seen as monolithic system, but rather as the end result of three systems:

Digital Library Management System defined as being the generic software that enables the creation and management of a digital library system.

Digital Library System defined as being the running software system that implements the digital library.

Digital Library defined as being the final system as perceived by end users.

As was previously mentioned, the basic premise that can be extracted from the DELOS digital library reference model is that a digital library can be seen from multiple domains. Domains are part of the Reference Model and are used to group core concepts of what should be provided (or served) by a digital library. The DELOS digital library reference model defines the following core domains (illustrated in Figure 2.2):

Content represents the actual information (both digital objects and additional metadata) kept and made available by the digital library. Usually sets of digital objects with common theme will be further grouped into collections.

User represents the communities that are served by and interact with the digital library. It should be noted that in this model the concept of user also includes those that have management duties within the digital library. Furthermore within user domain is also responsible for detailing the concepts that deal with access rights and system personalisation.

Functionality represents the services provided by a digital library to its users, be them end users or system administrators. The complete set of services that a digital library

provides should be tailored to the users it serves and to the digital objects required by those users. This domain provides the concepts required to model additional services that fall outside the core services implemented by nearly every digital library (digital object ingestion, search and retrieval).

Policy represents the concepts that covers the terms and conditions user under which the interactions between users and the digital library take place. This domain provides the definition of the concepts that go well beyond simple access management. It includes the agreements made with digital object suppliers, billing agreements or confidentiality agreements. The inclusion of this domain also ensures the support for the creation of new and evolving policies that were not in place when the digital library was first envisioned.

Quality represents the concepts that can be used to characterise and evaluate the behaviour of digital library. While some of the concepts covered by this domain are objective and can be measured automatically be the system, yet to take into account the human element that is also part of the digital library environment, there will be the need to define subjective evaluation criterion that can only be evaluated through direct enquiries to the users.

Architecture represents the actual software and hardware required to support the digital library. While the reference model does not require any particular software or hardware combination, it describes digital libraries as complex systems. Having a clear set of concepts that can be used both as a guide while building the digital library and as a platform to compare and ensure interoperability with the architecture of other digital libraries becomes an advantage in the long term.

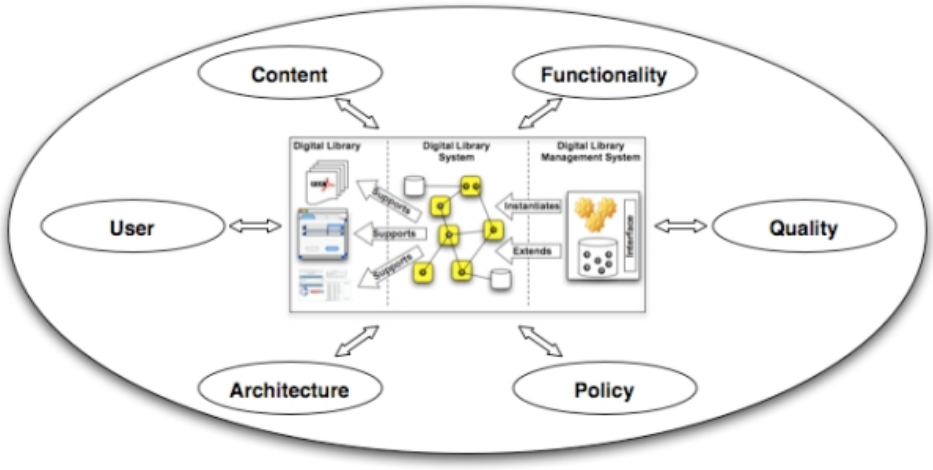


Figure 2.2: DELOS Core Domains [14]

In addition to these core domains, the DELOS digital library reference model defines a complementary domain, whose sub-domains while not part of the core model will need to be addressed by digital libraries. In this complementary domain we can encounter support for such concepts as Time (to represent time based concepts such as dates, periods or intervals), Space (to represent spatial based concepts such as positioning or physical location) and Language (to represent concepts inherent to formal written or spoken communication that might

be store in the digital library). The combination of the core and the complementary domains forms the base to further establish other concepts, thus defining an hierarchy of concepts, and properly describing the relations between concepts, both within a single domain and across multiple domains.

Just as a digital library can be seen in multiple perspectives, so can its users. According to the DELOS digital library reference model, users interacting with a digital library can assume one of four complementary roles (End User, Designer, System Administrator or Application Developer), where each role has a strong association with one of the three systems that compose a digital library. Given that user roles are bound to one of the three systems that compose a digital library and that said systems exist in a tiered hierarchical model, user roles also end up organised in a hierarchical structure. Furthermore, by explicitly defining the roles, the DELOS digital library reference model ensures that each role will share core concepts and vocabulary that will allow them to cooperate with users that are operating in a different role within the digital library without ambiguities. For instance, End Users interact primarily with the Digital Library, describing their issues and needs in terms of its model; Designers pick up those needs and requirements and adjusts the Digital Library model accordingly, changes that need to be relayed to System Administrators and Application Developers who can reflect those adjustments in the Digital Library System and Digital Library Management System parts of the model.

End Users are responsible for using the services made available by the digital library in order to submit, search and retrieve both primary digital objects as well as its metadata. Given this definition the role of end user in a digital library can be seen as an umbrella role that encompasses the role of the digital object creator, that supplies digital objects to the digital library, consumer, that uses the digital library to look for information and of librarian, that uses the system to catalogue and manage submitted digital objects.

Designers are responsible for designing and maintaining the information model that supports the digital library, taking into account the expectations requirements of the digital library's end users. As such designers are responsible for tuning the digital library system's parameters at the functional and content levels, deciding for example what query languages the system should accept, what classification schemes should be available to be applied to content, or even what third party repositories should be available to the end users of the digital library, effectively shaping the way the digital library works and is perceived by the end users.

System Administrators are responsible for selecting the software components that will support the digital library, and to configure them in such a way that they respect the Concrete Architecture defined for the digital library. With the responsibility to select the software components also comes the need to select appropriate hardware components, weighing the impact that each component will have when called to support the functionalities that Designers envisioned that can be provided to end users. This means that administrators should not be driven by purely economic criterion but by quality ones.

Application Developers are responsible for the actual development of the software that supports a digital library.

2.1.3 Streams, Structures, Spaces, Scenarios and Societies

Streams, Structures, Spaces, Scenarios and Societies (5S) [15] is another model for digital libraries. This model treats digital libraries as complex systems, with multiple possible definitions, whose complexity stems from the need to integrate within a framework multiple knowledge domains. The 5S model adopts a technological agnostic approach, based around a core formal model that describes the minimum set of components that must be present in order for a given system to be classified as a digital library. The formal model can be leveraged both to prove the correctness, completeness and consistency of the implementation of digital library, or to serve as formal underpinning for both new or existing digital libraries, guiding their design or evolution towards interoperability or even unification. The five "S"s (Streams, Structures, Spaces, Scenarios and Societies) that give name to the model are none other than the concepts (whose description will follow) upon which the core model relies in order to define a digital digital library.

Streams are a sequence of elements that form the basis of any type of content. A stream can be viewed from a static perspective where the relevance of the temporal-related information carried by the sequence is downplayed or simply ignored when interpreting the element sequence. In this perspective important information can be gathered from the structure encoded in the sequence of elements that compose the stream. This allows the representation of complex digital objects as sequence of simpler digital objects. For instance a multi-page text document can be seen as a stream of pages where each page is composed by a stream of characters. On the other hand a stream can also be seen from a dynamic perspective, where the temporal dimension of a sequence is used to represent the flow of information. In this perspective the elements that form the stream can be represented by clock times with an associated value. Both the clock time and the value is required in order to transform the stream into usable information (such as streamed audio, or a sequence of timed email messages).

Structures represent the formal organisation applied to the digital library. The organisation can be expressed through the use of ontologies, taxonomies, relationships or hypertexts. It should be noted that to cope with heterogeneity of content that might be part of a digital library's collection, some of it might actually be organised in a semi-structured format. This means that content still has some inherent structure, yet it is not a completely regular and rigid structure shared by every piece of content.

Spaces are sets of objects and the operations that can be performed on them while under well defined rules, a definition analogous to the mathematical definition of spaces. Being a very broad construct, it is suggested that when a part of a digital library can not be modelled by any other concept it should be modelled by a space.

Scenarios are set of states and events that trigger the transition between states. Scenarios can be used to describe the flow of information required to accomplish a given task, thus making them ideally suited to model for instance user workflows or the dataflows between the digital library's services.

Societies are composed by entities and their respective relationships. In this model, digital libraries arise to serve the needs of societies, thus making this the highest level concept within the 5S model. it should be noted in this model the term entity encompasses

anything that has any type of stake (be it direct or indirect) in the digital library, going from its users and developers, to the software and hardware that powers the digital library.

In addition to informal textual definition, the five concepts upon which the 5S model is based are also defined in formal notation, using mathematical constructions such as functions, graphs, or tuples. This means that the five core concepts of the model can be seen as low level formal blocks, that serve as base for the formal definition of other concepts that are commonly found associated with digital libraries such as digital objects, collections, repositories or services. The end result is a formal definition for digital libraries, in which a digital library is defined as the quadruple $(R, DM, Serv, Soc)$. In this quadruple, R is the repository that holds the content, DM is a set metadata catalogues, one for each collection held by the system, $Serv$ is the set of services provided by the digital library and that must include services for indexing, browsing and searching through the content and Soc is a society that is served by the digital library. This quadruple based definition only establishes the syntax through which the digital library can be represented, and not its internal semantics, constraints or rules, which are part of the internal definitions of quadruple individual components. This mathematical definition supports an ontology that can be used to describe digital libraries. Furthermore, the core model is designed to be extended in such a way that it can become a full fledge reference model [16].

While the 5S model was aimed at traditional digital libraries, that serve communities, one proposed extension augments it with the concepts required to be able to describe personal digital libraries [17]. This version of the 5S model is the only one from the "reference" models that recognizes that the thought processes, communication and information processing strategies of individuals are fundamentally different from those of communities. The 5 "S"s from the original model are redefined to better represent and deal with an individual's intrapersonal communication model, based on existing models presented in other studies [18, 19, 20]. In this new scenario, an individual's behaviour has to be intertwined with the digital library's functionality, which prompts the inclusion in the model of auxiliary concepts to support intrapersonal communication and thought processes. The inclusion of and individual's thought processes within the model means that personal digital libraries that follow this model are expected to be able to create workflows and provide services in such a way that mimics those thought processes, providing said individual with what he perceives as a more intuitive flow for information retrieval.

As with the original 5S model, this derivative provides a formal definition for a personal digital library. In it a personal digital library is defined by the quintuple $(Rcpt, WM, LTM, CNT, SOC)$, where $Rcpt$ is the receptor (a type of input service that interprets streams of user inputs, also called stimuli); WM is the working memory, a temporary repository for digital objects and stimuli relevant to the user current task (akin to the working memory in the intrapersonal communication models); LTM is the long term memory, the permanent repository of the user's digital objects and roles (also akin to the long term memory in the intrapersonal communication models), CNT is a controller, defined as being the triple $(finding, conceptualizing, reusing)$ that represents the ability to search through the personal digital library working and long term memory, while establishing the relation between collections of digital objects and the roles assigned to the user and recalling a set of previously existing scenarios that might be relevant to the current search; finally SOC is the society associated with the personal digital repository (from the 5S original model). the implementation of this formal

model for personal digital libraries has uncovered a set of challenges that should be tackled by personal digital libraries. These challenges include establishing what can be considered as relevant user behaviour and capture it; how to effectively use metadata to describe both personal information and user behaviour information; establishing appropriate conceptualizations, which include defining the relations between digital objects; how to design multi-cue find and retrieval services that can take advantage of both digital objects and of behavioural information available within the personal digital library and finally how to test the design of a new personal digital library (due to the lack of standardised well-known collections).

2.1.4 Functional Requirements for Bibliographic Records

While the previous models provide high level definitions and procedures useful for designing digital repositories, they do not establish how to organise the digital objects contained within the repository, since that task is highly dependent of the actual content that will be handled by the repository. Yet while low-level specific organisation is highly content (and context) dependent, there are several mid-level conceptual models that can be used to establish an initial hierarchy and generic relations between digital objects. One of these models is Functional Requirements for Bibliographic Records (FRBR) [21], a cataloguing schema developed originally for library collections that aims to provide an entity-relationship framework for relating data contained in bibliographical records to the needs of the users. Akin to higher level models in FRBR the definition of what constitute a user is not limited to end users of a library or library personnel, but also includes publishers, distribution agents and retailers among others. The FRBR model also attempts provide a framework generic enough that is able to accommodate various cataloguing schemas. Though the model takes into account the needs of multiple user types and attempts to establish a generic framework, it makes very explicit that it was designed with the needs of formal libraries (such as national libraries) in mind. Furthermore, the bibliographic record established within the model reflects some "*business rules*" that may not be appropriate (or even applicable) to every type of digital object or collections. In those cases, the model strongly suggests that the use of more suitable "business rules" or established traditions should take precedence over the ones encoded within FRBR, thus adapting the model to the current domain. The FRBR model defines that in order to a library to be able to cater the needs of its users, it needs to provide the tools that allow them to accomplish the following set of generic tasks (find, identify, select and access objects):

Find objects within the catalogue of a library that fit a given criterion provided by the user.

Identify objects within the library catalogue using data from a previous query, in such a way that ensures unequivocally that the objects described by the records are the ones wishes to access.

Select objects based on the provided records in such a way that ensures that the objects will be appropriate for the user's needs (for instance by choosing a book's version in the user's native language or a digital object that can be used with software available to that user.

Access objects based on data retrieved from bibliographic records, where in this context access can mean immediate access to object, placing a loan request or even be able to issue a purchase order (the last two scenarios are subjected to the object's availability).

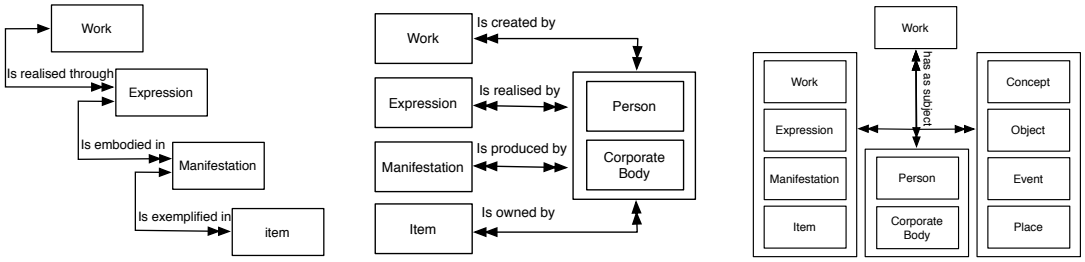
The key to successfully complete the previously described tasks rests on the data provided by the library's bibliographical record, which is simply defined as being a data aggregation. This aggregate is composed by the International Standard Bibliographic Description (ISBD) [22] elements, elements extracted from headings (that can be used in the creation of additional indexing structures), elements pertaining the internal classification schemes used on or within additional indexing structures or record sets, organisation elements used to manage and track the multiple copies of an object that might be part of a given library collection and annotations.

FRBR's model primary representation is a set of Entity-Relationship (ER) diagrams, complemented by the textual description of the attributed associated with each entity present in the ER diagram. FRBR uses a set of ER diagrams, instead of a single diagram, allowing it to clearly divide its entities into the following three conceptual groups (as seen in Figure 2.3).

Group 1 entities (work, expression, manifestation and item) describe the different points of view, ranging from a pure conceptual level to a physical level, from which a given object can be seen. The relationships between perspectives create a chain that can be used to link the original concept of an object to one of its possible concrete implementations.

Group 2 entities (person and corporate body) describe those actively involved in the production, dissemination and curation of objects, be them individuals or organised groups. Entities in this group establish relations with Group 1 entities, allowing the representation of roles or responsibilities assigned to Group 2 entities in the lifecycle of a Group 1 entity.

Group 3 entities (concept object, event place) serve as subject for objects. Group 3 entities can establish relations with entities of all groups, expanding the range of the entities that can be the target of a subject relation (thus allowing objects to have as subjects other existing objects or persons).



(a) Group 1 (b) Group 2 (c) Group 3

Figure 2.3: FRBR Entities [21]

The FRBR model defines a core set of information that should be present in a bibliographic record yet this core information set, and the model itself are flexible enough to be extended by other specifications. Two of the extensions made to the FRBR core model come in the form of the Functional Requirements for Authority Data (FRAD) [23] and the Functional Requirements for Subject Authority Data (FRSAD) [24]. FRAD extends the FRBR core model with the introduction of new entities, relationships and definitions, while at the same time it

expands and improves the definitions of the existing ones. The new entities, relationships and definitions in FRAD are intended to aid the representation of authority data, which simply put is another aggregate of information. While the bibliographic record is an aggregate of information about an object, authority data is an aggregate of information about one or more of the ten entities defined in the FRBR model. This new construct, when made available under a well defined identifier (or a well known name) can be used to establish a controlled access point over the underlying bibliographical records. An alternative way to create a controlled access point in this model is by aggregating other distinct controlled access points. As its name indicates, controlled access points serve as checkpoints that can be used to restrict who and how can access the underlying objects or records. Restrictions can be expressed in the form of rules, that might stem from the library itself or from external personnel or communities. As with FRAD, FRSAD also introduces new entities, relationships and their respective definitions to the core FRBR model. The new additions to the FRBR model are centred around Group 3 and are aimed at the representation of subject authority data. By providing improved tools to convey the relation between an object and its subject (i.e. the topics that are covered within a given object), FRSAD is attempting to create authoritative subject data that can be used to group multiple objects that share a common subject matter under the same subject designation, regardless of how the subject is named in the original object.

2.1.5 ABC Ontology and Model

The FRBR model and its extensions provide a model to support the cataloguing systems of traditional libraries. One of its underlying assumptions is that described objects and their attributes are stable and will not change significantly over a long periods of time. When an object suffers a significant evolution or revision, FRBR provides no way to represent the transition between states nor the time dimension associate with said transition. This makes the FRBR model unsuitable for use in scenarios where there is a need to establish and describe an object complete provenance to ensure its authenticity (such as archives) or where the objects being catalogued are volatile user-created digital objects that go through multiple iterative revisions that should be tracked. These are some of the issues tackled by the ABC Ontology and Model [25].

The ABC Ontology and Model has been designed specifically for digital scenarios, and its main goal is to provide a conceptual model that be used to ensure the interoperability between the metadata vocabularies of multiple domains. The first step to achieve its goal is to recognise that while "pidgin" metadata [26], such as Dublin Core (DC), is useful for many scenarios, the underlying formal principles and entities it encompasses may not be as defined as it would be desirable. This can lead to confusion when mapping entities from one vocabulary to another, that in turn become barriers to the creation of interoperable systems. The ABC model addresses this issue by proposing a set of basic entities that have the potential to be suited for use within multiple vocabularies and use them to construct a conceptual model (in the form of a "top-level" ontology) from which and to which more detailed ontologies can be mapped. This model also recognises that a given object set and their respective relations can have multiple representations with varying degrees of complexity, with the "more" representation being defined by the what is expected of the application that will use it. As such the model itself neither advocates nor rejects any representation type and is able to accommodate both simple (such as entity-property relationships) and complex (verbose properties-of-properties

approaches) representations, emphasising the ability of change between the two should the need arises. The model does this by establishing that a property is in fact a specialised form of a resource that can be associated with any other type of resource. Properties can in this way be "promoted" to first class objects with properties of their own. Classes in the ABC model can be broadly divided into three classes (temporarily, actuality and abstraction), being that within the ontology this division is achieved by representing the categories themselves as classes, from which more specialised subclasses can (and are) derived, and that in turn are themselves derived from the top level "Entity" class.

Temporality Category: classes in this category as used to express time and the effects it has in the properties of objects. This category marks temporal entities as first class objects, which enables the expression of complex interactions and states. In this context a state is used to associate time-sensitive properties to an object (for instance current or past ownership); an event is used to describe the transition from one state to another, with the associated temporal properties used to define event (and by extension state) boundaries; finally actions are used as verbs to detail the actual event that lead to the state transition. The use of states to deal with the time sensitive properties implies that there are time-insensitive properties. Time-insensitive properties are considered to be stateless and exist across time in the universes described with the ABC model.

Actuality Category: classes in this category are used to represent entities that are the embodiment or materialisation of ideas and concepts. These materialised entities have independent existence from the ideas or concepts that spawned them, and are composed of multiple facets some of them time-sensitive. Time-sensitive facets can be used to express temporal occurrences that affected a given object or entity, thus enabling the model to track that object's life cycle.

Abstraction Category: classes in this category are used to express concepts or ideas. Objects that fall under this category are stateless and do not have an independent existence. To exist they must be associated with at least one object or entity that belongs to the Actuality category.

The ABC model is expressive enough to allow, when needed, the description of complex universes, yet as it should be expected, taking advantage of them is itself a more complex task. A complex universe requires more time and effort to describe and eventually fill with appropriate (and preferably useful) data. Conversely, an intricate web of data, will require more elaborate (and potentially computationally more expensive) queries to take advantage of all the additional information available. These issues lead the model authors to issue a warning to communities seeking to build complex models to consider the costs and benefits associated with complex models and consider alternatives, such as supplying higher volume of simple metadata that can be extracted using automated processes.

Since the ABC model and ontology has been designed to be a top level ontology, it was never intended to be used directly, nor has among its target audiences the end users of digital libraries, or even the librarians themselves. Instead communities interested in developing their own ontologies, that can use guiding principles encoded within the ABC model as a template for their own ontologies, thus creating an indirect connection with the ABC model. Those communities that prefer more direct approaches can also choose to extend the provided ontology at key points. Furthermore, given its focus on interoperability, it can also be used

as the basis for a "*lingua-franca*", that facilitates the mapping of concepts across communities or vocabularies. In both scenarios the target audience are system builders and designers, not end users.

2.1.6 Europeana Data Model

The Europeana Data Model (EDM) [27] is the second data model used by Europeana to structure information. As an online cultural heritage portal, one of the driving forces behind the creation of this model was the need to integrate the requirements and standards of the several distinct communities (museums, archives, audiovisual) that contribute information to Europeana. The initial approach, called Europeana Semantic Elements (ESE), reduced information provided by the different communities to the lowest common denominator, coercing it into a reduced Dublin Core like form. this approach processed metadata supplied by the Europeana partners into the ESE form, discarding metadata afterwards, an approach with the potential to lead to unintentional loss of information due to the reduced expressiveness of the common form. The EDM is an answer to this perceived shortcoming of the original model. Its design was guided by three principles: the model must be able to cope with a multitude of data that comes from an uncontrolled open environment; the model must support a rich set of functionality and finally the model should reuse existing standard models and ontologies. In addition to those three design principles, EDM's design was also influenced by a set of seven requirements:

- Provide a clear distinction between existing physical objects and their digital representations.
- Provide a clear distinction between modelled objects and the metadata used to describe them.
- Allow the existence of multiple, possibly contradictory, metadata sets about a single object.
- Support complex object that are composed of other objects (part decomposition)
- Allow different levels of abstraction within an object's metadata.
- Provide a standardise metadata set, that can be further specialised if needed.
- Support an environment where the object's metadata is context-sensitive (i.e. it can come from different specialised controlled vocabularies according to the objects that is being described).

The conjunction of the design principles, established requirements and one of the project goals (Europeana aims to aggregate resources supplied by the project's institutional partners and enhance them by establishing relations between multiple resources, including publicly available resources from external entities) lead to the decision of endorsing the semantic web language family, Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL), as the data representation languages for the EDM. As an aggregation model, EDM is used to tie-up existing disparate data from multiple providers, enriching it in the process. When coupled with the choice of using the

semantic web language family for data representation, this strategy allows Europeana to actively participate in the Linked Data Initiative [28], not only by providing its own data but also (and more relevant for the Linked Data Initiative) by linking it to other relevant data sources.

At the core of the EDM model lie three classes, that form the basis for representing a given information package provided to Europeana. These core classes represent the object itself (*edm:ProvidedCHO*), a digital version of that object (*edm:WebResource*) and an aggregation (*ore:Aggregation*) that can be seen as the sum of the contributions of a provider to describe a given object. The first two classes can be seen as supporting one of the requirements previously defined for EDM, by providing a clear distinction between the object (which probably is what users are looking for) an its digital representation. The last class is used to tie both the object and its digital representation together, in one logical package. Furthermore it also demonstrates one of the driving principles of EDM, by reusing the existing Object Reuse and Exchange (ORE) vocabulary [29]. Descriptive metadata (i.e. metadata that describe actual properties of the objects) will be associated with the object itself (thus with instances of the *edm:ProvidedCHO* class), not with its digital representation. As a model, EDM maintains a neutral stance when it comes descriptive metadata. More specifically, it does not mandate neither an object-centric nor an event-centric approach, and instead provides supporting classes so that data supplying partners can decide the type of approach that better suits their data. It should be noted that the model supports the use of both approaches simultaneously, allowing the creation and coexistence of simple data-driven views, more intricate event-driven views and everything in between. Some of the descriptive metadata, while also related with the object, might actually be describing other resource or object. In order to enrich the model, EDM provides a set of classes that can be used to model these implicit context-dependent entities (agents, places, time spans, events, etc). Submitters are therefore encouraged to provide richer contextual metadata, even if that means using their own controlled vocabularies or ontologies to represent it.

EDM was developed with aggregation of existing resources in mind. During this process it is highly probable that it will receive information about a given object from multiple partners, representing an object from different perspectives. The model must therefore have a way to manage both sets of data, even if they contradict each other. Instead of coercing the both datasets into a single view, EDM re-purposes the concept of proxies (from ORE) to solve this issue. In this scenario each dataset is a contribution from a different partner and therefore forms the basis of an aggregation. Each aggregation will then have a proxy, to which the respective descriptive metadata will be attached in lieu of the attaching it to the actual object. The object being described is still part of both aggregations (which can now be asserted as being the same object described from two distinct perspectives). While it can be tempting to directly expose the proxy objects (since they can be associated with an object's descriptive metadata), it is highly unlikely that users know beforehand what are the objects they are standing in for, so EDM adopts the strategy of also directly linking the proxy to the object, and exposing the object. The overall proxy strategy can also be used to maintain within the model the original descriptive metadata supplied by Europeana partners, separating it from standardised metadata that might automatically be added to the model for interoperability purposes (for instance when mapping from a partner's controlled vocabulary to a more widely used vocabulary).

2.1.7 CIDOC Conceptual Reference Model

CIDOC/CRM [30] defines an ontology for cultural heritage information. Cultural heritage is composed of both information and physical artefacts that can tell the story of both people and the environment where they live. The goal of CIDOC/CRM is to provide a common framework that can be used to exchange information between institutions of the "cultural heritage sector", promoting interoperability in an attempt to transform localised information sources into a global resources. It does this by defining the precise meaning of common elements, and establishing the relations between them in a formal ontology. As with other ontologies, the one defined by CIDOC/CRM covers only core concepts, and makes no assumptions over the underlying data, and the specific terminology required to support it. Specific terminology has to be discussed and defined within the target community that intend to use the CIDOC/CRM ontology. This means that the model does not attempt to model what should documented, but rather the logic of how it is documented. Furthermore it acknowledges that specific implementations might turn relations that are explicitly represented in the model into implicit relations due to technological constraints or optimisations. CIDOC/CRM has two root classes: Entity and Primitive Value. This first serves as an umbrella class for everything that can be described by the CIDOC/CRM model, except for the raw values of properties associated with them. These are generically represented by the Primitive Value class, which ultimately needs to be subclassed (or even replaced) by native values and their respective units when in use. Although CIDOC/CRM defines 90 classes, the core of the model can actually be understood by looking at the definitions and interactions of 7 classes (illustrated in Figure 2.4):

Temporal Entities is the top level class that serves as root for the representation of events (it should be noted that CIDOC/CRM is an event-centric model). Events are used to bind time to other types of objects which means that regardless of the object type, if it has a temporal dimension that needs to be represented, that representation will be done through the use of an event.

Time Span is the top level class responsible for representing abstract temporal intervals, characterised by (potentially having) a beginning, an end and a duration. Subclasses should be used to add further (and more precise) meaning to the periods of time that their are describing. Time Span provides an approximation to time that allows some of its components to be actually be unknown, while still retaining the ability to represent that two events occurred simultaneously (as far as the available knowledge goes) by linking them to the same instance of a Time Span.

Places is the top level class that represents the space (in the mathematical sense, not necessarily in the physical world sense) where an event takes place. By adopting a more mathematical approach, the Places class can be used to represent both real world locations and imaginary ones (such as locations mentioned within a book), thus giving an added layer of flexibility to the model.

Actors is the top level class that represents anything that can take any type of legal responsibility when they participate in an event, and whose actions have an effect on other objects, affecting them in some way. Its subclasses can be used to represent individuals and groups (be them formal organisations or informal groups).

Conceptual Objects is the top level class for objects that are the product of the human mind. Conceptual Objects require vessels (be them a sheet of paper of the human mind) to exist. Unlike the their vessels, Conceptual Objects can not be destroyed (since that is what would happen to the vessels that carry them) but can be instead lost. This allows the representation of ideals that rise and fall throughout the course of time as well as the representation of artwork or stories that we know once existed but whose vessels have been permanently destroyed thus consigning them to oblivion.

Physical Thing is the top level representation for objects that have a physical presence. It is used to represent physical objects that can serve as vessels for Conceptual Objects and with which Actors can interact, with the result of the interaction constituting an event. A Physical Thing can be destroyed, which in this model means that it is transformed in such a way that it no longer relevant to keep representing it as is or to track what happens to the products that resulted from said transformation.

Appellations is the top level class that represents names that can be given to entities. By treating the names given to entities as first class objects it becomes possible to see and describe the history of the name itself, including its usage throughout time, as well as allowing the model to differentiate between a thing (that has its own identity) and the (possibly) many names that can be attached to it across temporal and cultural dimensions. As a result of this last trait, the names represented by instances of this class (or subclasses) do not carry any semantic weight by themselves, carrying instead the weight of tradition or convention.

Types are a specialization of Conceptual Objects intended to represent the classification that can be attributed to a given object. An object can have multiple types associated to it, being that those types can come from multiple classification schemes. The types themselves can serve as basis for further subclassing in order to further flesh out and refine classification hierarchies. Analogous to what happens with Appellation, considering Types to be first class objects allows the Types themselves to be tracked across both temporal and cultural dimensions, but more importantly, it allows them to be represented alongside the objects they are meant to classify, effectively encoding the entire classification scheme within the data itself, a feature useful for instance in digital preservation scenarios.

These classes are linked together by several types of relationships. in spite of the multiplicity of relationships provided by the model, the relationships themselves usually fall within one of the seven broad relationship categories: Identification, Observation/Classification, Part-decomposition, Participation, Location, Influence and Reference. Identification relationships links a given object with one of the appellations (names) by which the object can be known, supporting the model's view that everything can have a name associated with it. Observation/Classification links a given object to the terms and types that can be used to classify it, supporting the model's view that everything that can be represented within the model can be classified in a myriad of ways. Part-decomposition links a given object with its constituent parts, allowing the model to describe both composite items and their individual elements. It should be noted that Part-decomposition can be applied not only to objects, but also to temporal periods, places and even actors (tough when describing actors it usually refers to groups of actors and not to the individual body parts of actors, though this may vary according to

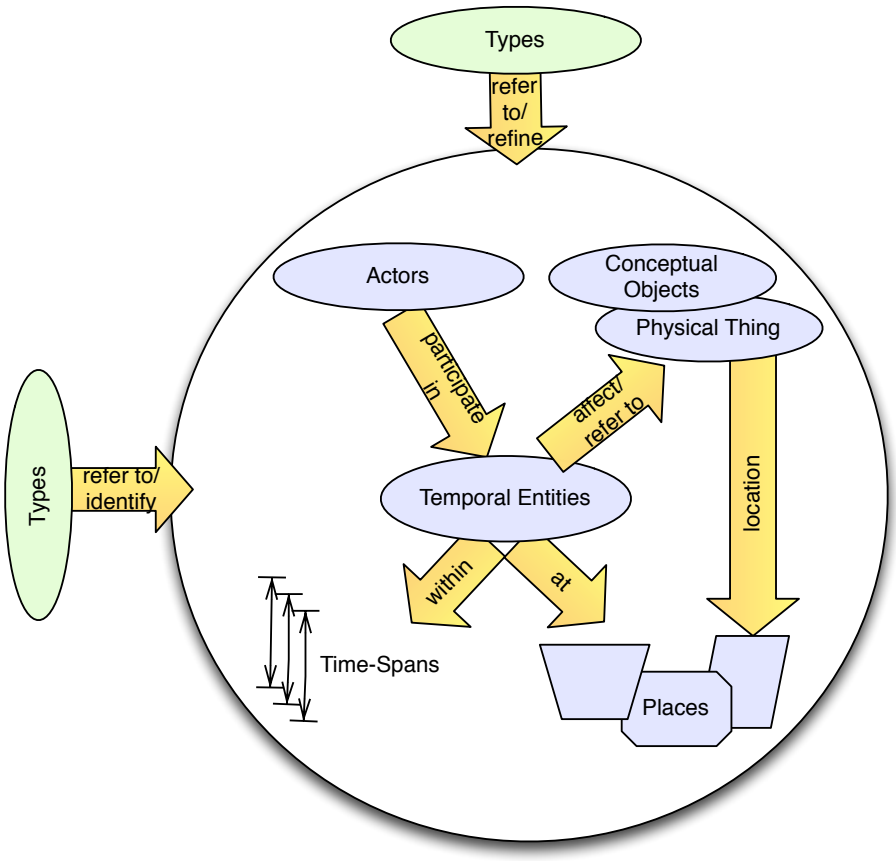


Figure 2.4: CIDOC Top Level Classes [31]

the model's surrounding context). Participation links a given object to the temporal entities in which it participates, allowing the model to describe a given object's journey through time. Objects involved in Participation relationships must be persistent items i.e. objects that maintain their identity over time allowing observers to identify them as being the same, without the need of constantly observing said object throughout its entire existence. Location links a given object to a location, where both the object and the location can be either physical or conceptual objects (for instance physical objects, actors, events or time periods). Bounding time periods to a location allows the model to describe cultural phenomenons more accurately, enabling the representation of the effects and impacts of global phenomenon as perceived in a specific location (that due to propagation delays might not have the same starting time or last as long as the global phenomenon). Influence links two objects, where the presence of one can be said to be determinant to the outcome or development of the other. This type of relationship allows a loose representation of, for instance, of how a given event influenced the development of a given idea. Finally Reference relations allows objects to describe and reference other existing objects, being that the primary use of this relationship type is to link conceptual objects with their real world vessels or representations.

CIDOC/CRM's role as an umbrella ontology capable of promoting system interoperability is cemented by its status as a standard (ISO standard ISO 21127:2006) and by the multiple

mappings it provides for other ontologies, vocabularies and models. Relevant vocabularies and models that are either mapped or extend the CIDOC/CRM include the Dublin Core Element Set [32, 33], the ABC Ontology and Model, FRBR via the FRBR - object oriented ($FRBR_{oo}$) specification (developed as an alignment of the concepts of both models, with the resulting specification being a specialisation of CIDOC/CRM) and the Europeana Data Model (which incorporates concepts from CIDOC/CRM, and that can be mapped to it by way of the existing mappings for $FRBR_{oo}$ and DC). Albeit complex, its versatility and existing mappings make it a good choice as starting point for the development of more specific ontologies in the cultural heritage area. An example of this can be found in [34], with an proposed extension to CIDOC/CRM to describer in more detail the provenance information of a given object.

2.2 Tools

2.2.1 Digital Repository and Digital Library Software

The management of collections of digital objects is usually entrusted to specialised software. Digital Repository Software (that should not be confused with the Digital Repository itself) offers a platform upon which several types of repositories can be built. One of the more popular use cases for this software class is its deployment in institutional scenarios, where it serves as the backbone of institutional repositories. Institutional repositories gather in a single location all the content generated within a given institution. Depending on the institution goals, gathered content can be made available in its entirety, (or partially) to the general public, or remain available only within the institution. In some scenarios, particularly academic ones, gathering content at a central, publicly accessible repository can serve as a public relations strategy to increase the visibility of the that institution's research output. Gathering the collective output of an institution can also help its efforts of digital preservation, be them guided by legal obligations to maintain and curate produced materials for a predetermined period of time or simply by the desire to keep an accurate record of its history.

There are several digital repository software platforms, both commercial and open source, and ranging from ready to deploy packages such as ePrints [35] and DSpace [36] to more complex solutions designed to serve as repository building blocks such as Fedora [37]. Data from the mash-up site Repository 66 [38] appear to point out that the more widely deployed platforms are open-source platforms. DSpace in particular has been chosen as the underlying digital repository software in a dissertation that assessed the viability of establishing repositories "intended for the use of the average individual for the preservation of personal digital objects" [39].

DSpace was selected as the base for a personal digital repository due to its nature as a ready to use package, with its fully customisable web interface singled out as "a comfortable and convenient means of interaction for the typical individual". A DSpace repository is organised into communities, with each community potentially having multiple item collections. In the work Lesley L. Peterson [39], the DSpace concept of communities is re-purposed to represent family units. Each family member is assigned at least one collection for which he becomes the sole responsible, which in this case means that on one hand that family member has complete control over the collection's content as well as the collection's overall theme and organisation (within the limits of DSpace) on the other hand, it will be responsible for choosing whom within the family has access to the collection (or to each individual item). In addition to collections managed by the individual members of a family there is also an (optional) shared family

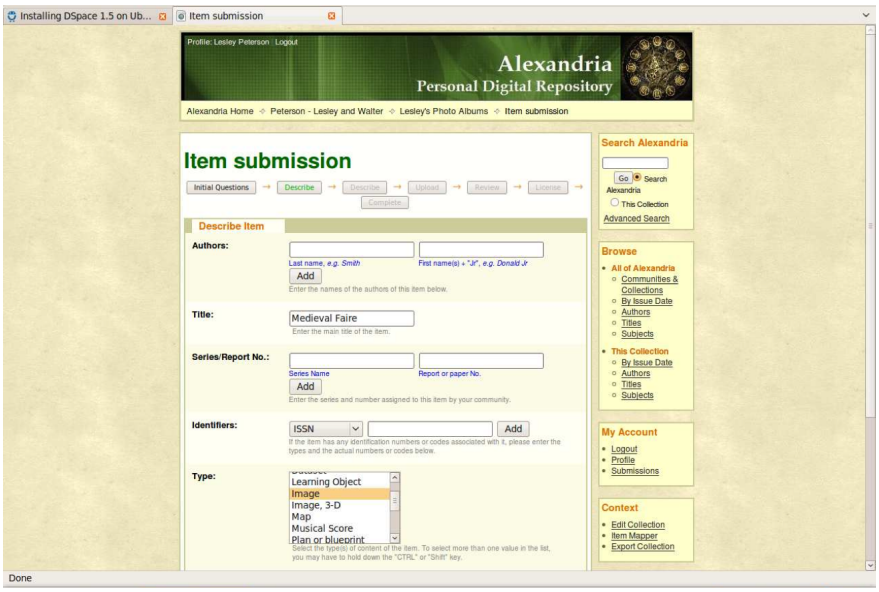


Figure 2.5: Reskined DSpace Object Ingestion Workflow [39]

collection to which every member of the family can contribute either with content from their own collections or with exclusive content. With this scheme, a single DSpace instance could be used as the underlying repository for the digital objects of multiple families. In addition to re-purposing the the community concept, this work also introduced modifications to DSpace's user interface and suggested that in order to elevate the produced prototype to commercial software grade DSpace would have to be modified to support automated management of group membership/permissions, which in the presented prototype needed to be managed and maintained by an administrator. Though the aim of the work was to produce a repository that could be used by "typical individuals", even with the proposed modifications in place it is doubtful that the resulting repository could be completely managed, maintained or even used just by "typical individuals". This criticism is supported by the fact that there was no no mention of any type of modification to the content ingestion workflow. Even though it sports a customised interface, the ingestion workflow is still the default multi-step process provided by the DSpace core designed to be used by trained personnel when working with carefully selected content. Maintaining the default ingestion process means that "typical individuals" need to fill several fields worth of information about each individual object, as seen in Figure 2.5, albeit some of those fields may not even be relevant for some types of objects, for instance International Standard Serial Number (ISSN) when dealing with an images.

In contrast with the (nearly) ready to run solution offered by DSpace, Fedora opts by providing an architecture and core services that can be used as a foundation for digital repositories. This approach ensures that Fedora based repositories can be completely customised to their user's needs, and those needs can go beyond the requirement to place institutional (or corporate) branding in a web interface, up to the point where entire workflows, services and data models must be developed from the ground up to ensure the resulting repository will be suitable for the community it will serve. Furthermore, while the standard installation comes with multiple tools (that range from command line utilities to a flex based web interface) geared for administrative tasks, their focus on system administration makes them ill suited

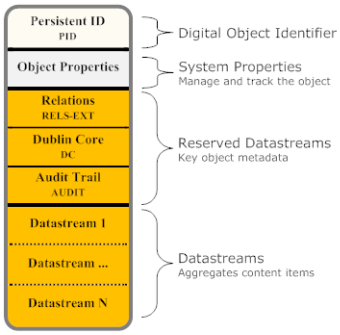


Figure 2.6: Fedora Digital Object Structure [42]

to serve as end-user interface, which is usually provided by some other digital asset management tools such as a Fedora aware Drupal instance (be it through the use of Islandora [40] or by a custom plugin that invokes the repository's Representational State Transfer (REST) Application Program Interface (API) [41]). While on a superficial analysis this approach can appear to be disadvantageous (after all, who wants to have extra work if there are ready to use solutions available) it also serves to showcase one of Fedora's strengths: the flexibility to adapt itself in order to accommodate specific needs. This flexibility is not only present in customised workflows or in the deployment of specialised services to complement the repository but also in the underlying object model that supports the repository itself.

The flexible object model (seen in Figure 2.6) allows Fedora to capture in detail both digital objects and the arbitrary relations between them, which stands in contrast with ready to deploy solutions (such as DSpace) that usually rely on fixed field sets (often based on Dublin Core) and contributes directly to Fedora's ability to assemble and represent complex universes. Aligned with seminal definitions of what is a digital object (such as the one given in [43]), in Fedora's object model a digital object is not simply the bitstream that represents the object's content, but a composite entity represented in an Extensible Markup Language (XML) structure that bundles together a persistent identifier, the system properties required to managed the digital object and one or more datastreams. A datastream is itself a complex entity that is composed by content and descriptive attributes. Datastream content can be encapsulated within the datastream itself, physically stored in the repository or even be held at an external location, while the datastream attributes include (besides "standard" attributes such as identifier, creation and modification dates, checksum, etc) mime type and alternate format identifiers (for the datastream content), alternate identifiers that can be associated with the datastream, such as an Digital Object Identifier (DOI), and the control group that describes how to access the content itself. This last attribute, control group, has the following four valid values:

Internal XML Content signals that the datastream content is stored inline with the digital object.

Managed Content signals that the datastream content is stored within the repository and its content location element is given by either an Uniform Resource Locator (URL) or an internal repository identifier.

External Referenced Content signals that the datastream content is stored in an external

location not managed by the repository, its content location element will be an URL, yet the repository is still responsible for mediating access to the content.

Redirect Referenced Content signals that the datastream content is stored in an external location, its content location element will be an URL and that the repository should not mediate the access to the content, and instead should use the redirect the client to content.

Most datastreams in Fedora essentially represent different versions of the digital object (a digital object that represents an image can include a datastream for the image itself and another for a thumbnail version of the same image) and are therefore treated as black boxes, essentially shifting the burden of interpreting the datastream content to the application that needs to access said content. Nevertheless Fedora defines four special datastreams that is able to use itself: one for the digital object's metadata based on Dublin Core and created automatically if not provided during the content ingestion process; one system managed (i.e. that can not be edited manually) for the digital object's audit trail and two datastreams (Relations External and Relations Internal) dedicated to describe relations. The Relations External (RELS-EXT) datastream is used to describe relations between digital object (be them directly managed by Fedora or external objects referenced by an URL). Relations defined in this datastream are interpreted as going from the current object (i.e. the one that contains the RELS-EXT datastream) to the other object; inverse relations, if any, have to be defined in the corresponding RELS-EXT datastream of the other participating object. The Relations Internal (RELS-INT) datastream is used to describe relations between the datastreams that compose the digital object. Both of these datastreams use the RDF-XML syntax to express the relationships being asserted, with subjects being defined using *<rdf:Description>* elements (which in the particular case of the RELS-EXT datastream can only appear once). Relationships can be expressed using either Fedora's own vocabulary or external vocabularies, though it should be noted that the use of Dublin Core properties within these datastreams is strictly forbidden, since Fedora already provides a reserved datastream for it. Another restriction placed upon these datastreams is that unlike the "*regular*" RDF-XML syntax, Fedora forbids the use of nested assertions. After ensuring that established relations fulfil these (and some others) rules, they become part of the datastream, indexed and inserted into a triple store where they become part of the global graph that encompasses the entire repository. Fedora exposes this global graph via a specialised web service which allows the graph to be queried using for instance SPARQL Protocol and RDF Query Language (SPARQL) [44].

When we take into account the object model, with its multiple datastreams that can be used to provide both multiple perspectives about a given digital object and support for intra and inter-object relations, and its service based architecture, both traits that support the claim that Fedora can be adapted to support nearly any model within the digital repository universe, it might appear that it could also easily be adapted to serve as the core of a personal digital repository. Yet in this regard, it is its own flexibility and depth that works against it, since these traits are attained at the cost of increased deployment, administration and customisation effort. This effort must be maintained throughout the life cycle of the repository in order to ensure that it serves the needs of its users. Keeping up with a moving target such as personal interests would require either a very broad (and consequentially shallow) approach in the way they are modelled, or constant adaptations throughout the entire repository stack, a process that is unsustainable if it has to be done by hand for each customised deployment of a Fedora backed repository.

In its essence, Fedora is a canvas upon which different kinds of repositories can be built. Though it didn't met all the requirements, its extensibility and (partial) compliance with the OAIS model has led it to be selected as the base for one of the prototype digital archive system developed during the Paradigm project [45]. Paradigm stands for Personal Archives Accessible in Digital Media, a pilot project that attempted to bridge the gap between physical and digital archiving techniques when it comes to dealing with personal content in the possession of and produced by individuals of interest, such as politicians or scientists. Paradigm encompassed three pilot projects that created archive systems for the materials collected from of active politicians, Internet navigation (of pages belonging or associated with relevant individuals) and from a traditional physical archive collection that included among the deposited materials old computers and disks (an activity dubbed "*digital archaeology*"). These pilot projects resulted in a series of recommendations [45] and a workbook with best practices for archivists [46]. Arguably one of the most important pieces of insight that arose from this project is that "*[t]he nature of digital environment requires creators to become curators in their own right*", going so far a dedicating a chapter of the workbook with generic tips that content producers can use to organise their content before handing it over to archivists. While everyone can benefit from the "tips" offered in the workbook, Paradigm as a project heavily relies on formal archives and archivists. As part of the final project report, Paradigm advocates the use of pro active content gathering strategies that put archivists in contact with content content creators early in the life cycle of the content they wish to gather and archive. While this type of approach might be feasible for some scenarios, where it is possible to determine that an individual will produce relevant content such as in their active politicians scenarios (and even there for various reasons they ended up working with the staff instead of directly with the individual in question), it is impractical to extend it to the general population (as it is impossible to assign a personal archivist to each individual), leading to recommendation 21 [45], that essentially calls for the research and development of systems that (albeit provided by memory institutions) could be used by individuals to create and manage their own archives.

On the digital library side, existing projects take distinct approaches when dealing with the user generated content that would normally constitute the bulk of the content placed in a personal digital library. On one end of the spectrum we can find JeromeDL [47], a digital library that takes advantage of both semantic and social network technologies in order to deliver a distinct experience to each user. It uses a simplified model, seen in Figure 2.7, to organise bibliographical references, augmented with the use of specialised ontologies MarcOnt [48] or Friend Of A Friend (FOAF) [49]. What distinguishes JeromeDL from other digital libraries is its focus on user interaction, both between users as well as between users and the library content. JeromeDL allows its users to create personalised collections, which it dubs of personal digital libraries, whose content comes from the main digital library. Users are free to share their personal digital libraries with other users, and to manage their contents by adding annotation to "their" content and assigning it to personalised categories. Given that the personalised categories of one user can be linked to the personalised categories of other users, this mechanism effectively becomes a method to generate both communities that share the same interests as well as community driven folksonomies. Furthermore, both the underlying ontologies as well as the established folksonomies can be used as vocabularies in semantic queries that span the entire digital library and are available both to users and to computer agents. The end result is the creation of a digital library system that while not handling user generated content, it guides users to additional content that they might be interested in consulting (akin to a recommendation system), based on semantic relations established by

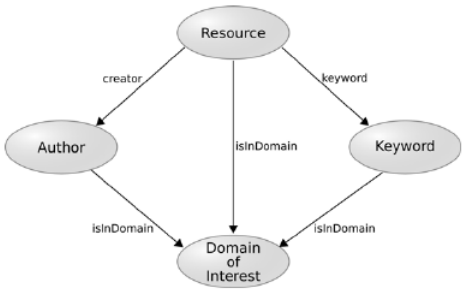


Figure 2.7: JeromeDL Core Model [47].

both they system itself as well as its users, eventually allowing users to discover content that would otherwise be unnoticed (for instance because it was written in a different language than the one used to perform the search).

PDLib [50] is a project that also aims to provide each user with a personal digital library. Unlike JeromeDL, where the content of its users personal digital libraries stem directly from the content available in the digital library system itself, in PDLib users can supply their own content, thus allowing more variety when building their own personal digital libraries. User generated content is organised in collections that behave in roughly the same way as folders in file system. Users are expected to manually select an appropriate metadata scheme among the ones that PDLib supports, as well as manually fill its fields when submitting their own content. For the sake of interoperability, PDLib aggregates metadata from each of its users' personal digital libraries [51], exposing the aggregations to the outside through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [52]. As with JeromeDL, users can choose to share parts of their personal digital library with other users of the PDLib system or alternatively, users can directly share content with external users (i.e. those who are not registered in the system) via email. It should also be noted that when users search the PDLib system, the delivered results can (optionally) come from the personal digital libraries of other users to which they have been granted access as well as from some external institutional digital libraries, in addition to the search results that come from their own personal digital library. A distinctive trait of PDLib is that is was designed from the ground up to allow access from multiple platforms. This means that the system can be accessed through the use of web based clients or specialised clients, being that the specialised clients were primarily created to support access from mobile devices. These specialised clients are are responsible for displaying a user's personal digital library in a format more suitable to mobile devices taking into account restrictions placed upon screen size, available processing power or bandwidth. While PDLib users can use the system to manage their own content and even share it, PDLib does not provide any way to describe that a given piece of content is related with another piece of content. PDLib users are also limited to the built in metadata vocabularies provided by the system and can neither define new fields nor import existing metadata vocabularies, while support for content versions is also missing from the PDLib specification, with new versions simply overwriting previous versions if assigned to the same collection. These issues hinder the ability of PDLib to support complex models, particularly if one's intention (or need) is to provide as much context as possible through intricate relations to the content stored in the system.

A different take on personal digital libraries is offered by Universal Personal Digital Library

(UpLib) [53, 54], a project that aims to provide a digital replacement for traditional physical personal archives. While sharing the mission of ensuring access to content with other digital libraries, UpLib does not claim ownership of the content it manages and instead it explicitly states that content should remain accessible even when UpLib is not running. Though it comes with a web interface, UpLib is supposed to be running on a local computer (as opposed to being a service accessed on a remote system) and goes as far as informing the user where the content that it is managing is located within the local filesystem. As a digital replacement for traditional paper archives, UpLib offers its users services and workflows that are able to transform paper documents into their digital equivalents. This process begins with a scan of the document, that will produce one or more image files, which are then fed to multiple Optical Character Recognition (OCR) modules for automated text recognition and extraction. The physical acquisition workflow is followed by a workflow shared between born digital and acquired content, designed to generate different representations (called projections) of the ingested content and to extract metadata from the content itself. Projections are used to feed the web interface, providing both visual and textual clues to help users locate content. The web interface by default displays thumbnails of the held documents in most recently added order with additional information provided by tooltips. UpLib also provides a document reader application (that can be used as a browser applet and as a standalone application) called ReadUp that takes advantage of the previously generated projections to display the personal digital library content, allows users highlight or add notes (textual, image or even freehand scribbles) to content, with the added benefit of user generated metacontent being permanent, living in the personal digital library as part of the content's metadata projection. Being so tailored to replace paper archives, UpLib's existing workflow is a not a good match to process content that does not have a direct paper equivalent, such as audio or video (though it supports them). Though this issue can be mitigated, since UpLib is designed to be extensible, at the time extensions need to be created by the users for their particular use cases, which is not an optimal approach when dealing with personal scenarios.

While the previous approaches offered its users the possibility to deal with (nearly) arbitrary content types that can possibly cover multiple subjects, Apollo [55] focused its efforts in supporting just one subject: music. Apollo acts as a cross between a personal digital library and a sketchpad upon which musicians can freely place fragments of melodies, text or images to support their creative process. It was designed by taking into account the input of musicians, in the form of diary studies, where participating musicians used a provided physical notepad to keep track of their creative process, which were supplemented with interviews to the participating musicians about their daily creative activities [56]. From these, the authors were able to define the basic features needed by their software in order to better support musicians. For instances while the initial vision might have included the use of Apollo "on-the-go" to capture everything in real time, the interviews showed that its prospective users already had schemes in place to do that and that instead they would be more interested in a system they could use to integrate multiple disparate elements. In Apollo metadata is used nearly only as a search aid and its extraction and production process is completely automated, with the only exception to this being metadata created by the users during the creative process, in the form of comments or annotations, which can be attached to any element in the interface. The system itself is built around the concepts of hypermedia and frames, where any type of content (be it text, images, sound snippets our entire documents) can be placed inside frames which can be linked. This approach establishes a navigation hierarchy, with parent frames that have links to children frames. Search can be used to quickly locate and navigate between

frames and supports both text and audio queries. After testing, users felt that one of the main advantages of Apollo was that it as a tightly integrated application that freed them from having to juggle between multiple distinct applications (for instance a digital audio application for sound edition and a text editor to take notes), which can be distracting during the creative process. While completely focused on the musical creation process, Apollo demonstrates that it is possible to capture the context that surrounds said creative process, by managing within the same environment disparate content types that provide additional insight to how the final result has evolved. In contrast PhotoGeo [57] goes on the opposite direction, by managing only a single content type that may cover a myriad of subjects. PhotoGeo proposes the creation of a photo-centric digital library that enhances the photos it stores with additional metadata, such as when, where or who was present when a photo was taken. PhotoGeo relies on a client for mobile devices that acts as an enhanced camera application, that in addition to taking the photos also suggests annotations and submits photos to the digital library. Annotation suggestions are generated by two algorithms, one dedicated to detect the event that the photo depicts, which is responsible for the generation of temporal and spatial information, and one dedicated to identify who appears in the photo. In order to increase the accuracy of the suggested annotations, the algorithms take into account additional information gathered from online calendars, friend location services or geographical reference systems. If the mobile client does not have network access it runs in the device a modified version of the algorithms, that in addition to be greatly simplified does not take into account additional information sources, while if it has network access it takes advantage of the digital library infrastructure to run the complete version of the algorithms, that may integrate information from additional sources (if the information is available). The end result of this process is a suggestion list with annotations that identify who, where and what was happening when the photo was taken. Interaction with the digital library to retrieve stored photos is done through a web interface. Annotations previously placed into stored photos play a key role in the search mechanisms exposed by the digital library, allowing users to create custom field/operation filters, allowing users to query the system by standard metadata (file name, camera manufacturer and model, etc) as well as by the provided annotations (for instance, query by distance to a know reference point or by precise location, by general description of the depicted event, etc.). They also contribute to the creation of spatial and temporal based visualisations for the photo sets stored in the digital library. In the end this project highlights the importance of information integration. By drawing information from multiple sources when there is network access, the algorithms provided are able to generate better annotation suggestions, which increases the end user acceptance rate of the suggestions and reduce the number of corrections that they need to do.

2.2.2 Other Tools

While digital repositories and libraries place a premium on the formal categorisation and organisation of collected content, placing them on the realm of institutional tools, other types of tools have different focuses and are consequently aimed at other audiences. Backup tools are an example of the existing diversity of approaches, with its sole focus being the creation of exact and (when possible) redundant copies of whatever content its users have lying around. This type of tools can be useful in both personal and institutional scenarios, with the target audience for each tool being defined by its price and features. They come in a variety of complexities ranging from cobbled-together scripts that copy individual pieces of content to a

remote server to integrated solutions that create snapshots of an entire storage volume or even dedicated solutions for specific devices (such as mobile devices). Each existing approach has its own upsides and downsides, yet their shared focus on the bitstream might be detrimental in the long run since it implicitly trusts users to do significantly more work than what they might expect. Users are expected to keep track of what piece of content went to each backup, to ensure that backups are physically distributed through multiple locations and/or media types, to come up with their own organisation schemes (most of the times expressed through whatever folder scheme is in place at backup time) and remembering the context which surrounds each individual piece of content. Furthermore, even though the bitstream of a given content piece might be intact in one of the backups, it might be unusable if that (or another) backup doesn't also contain everything required to actually use that content. This last issue is addressed by Home and Personal Persistent Long term Archiving (HOPPLA) [58], which is a digital preservation tool that initially presents itself to users as a backup tool with a minimal graphical interface. It can create backups of content from local file systems, remote servers accessible by Secure Shell Protocol (SSH) and (in the first departure from typical backup programs) from email accounts, with the generated backups being placed either on a different local disk from where the original content came or on remote servers. Another way in which HOPPLA distinguishes itself from a regular backup tool is that it keeps track of the archived content, a feature that opens new options to deal with said content. HOPPLA is capable of detecting that a a given content piece is actually a new version of previously archived content, allowing users to track content versions (and their modifications) throughout time. Furthermore, not all content is taken as face value, with certain content types being recognised by the tool itself. Metadata is extracted from recognised content types and the tool offers to the user the option to perform format migrations when the content is, for instance, in a format that is no longer in widespread use. Format migrations are performed locally, so no potentially sensitive content is sent to outside servers. Despite having an automated format migration workflow, format migrations are not performed blindly and are governed by user adjustable preservation levels [59]. Preservation levels control what actions are performed over a given content type, with a lower level meaning that less actions should be performed and conversely a higher level meaning that more actions should be taken on a given content piece. The only part of the tool that is not local is the one that deal with rule definitions. To lower the overall complexity of the system rules are not user definable, being instead defined and maintained by an external group of experts, kept on an external server and accessed by the tool on demand. Rules underpin the preservation level system, and are used to decide what how a perform a certain action with the tools available in the system. They form both its most interesting asset as well as its achilles heel. On one hand, since rules are created and maintained by outside experts, users are only required to have rudimentary notions about digital preservation (which in this case means just spreading out the backups and using HOPPLA), on the other hand rules created in this way need to be as generic as possible so that they can be applied to multiple pieces of content and thus are not tailored to the specific needs of a user or even of a collection.

In addition to generic backup tools that target local content, there are also specialised backup solutions for online content, particularly for social network content. These tools are usually part of the services offered by the online platforms (though there are also third party tools) under various pretexts, ranging from the creation of personal offline backups to ensuring data portability (both to and from those services). What comes out from those tools varies according to the service and ranges from self contained javascript driven local web application

with search capabilities as in the case of Twitter [60], to a bare-bones static HyperText Markup Language (HTML) version of the profile, timeline and other staples functionalities as in the case of Facebook [61] or to a collection of service dependent files designed to be imported into other applications in the case of Google Takeout [62] (an umbrella service that covers data in the majority of their available services). In the first two services, the supplied content is (nearly) self contained and is useful for little more than to serve as an offline backup in case the service ever ceases to exist, since there is no way to use the received backup to restore in the service, an accidentally deleted content piece or even an entire account (if the user closed its account and request at the time the removal of all existing data). Another issue with the first two services is that users are unable to select what they want to backup, with the option always being between creating a backup of everything or not creating a backup at all, regardless of if they have previously requested to receive a backup. This eliminates the possibility of performing incremental backups, a feature that could be important for users with large amounts of "heavy" content such as photos and videos and the possibility of creating selective backups of just part of the data (for instance a specific photo collection or comment thread). These issues are absent from Google's backup services, yet their implementation is not fault free. While users are able to select which content they want to backup or restore, and can do it on a per-service basis (or for all supported services at one), when they choose to actually perform a backup the consistency of what they receive varies wildly from service to service. While this is in part expected, given the different nature of the services and the emphasis placed into the backup tool as a data portability tool, it will require additional effort from the user (for example locating the applications that can deal with the format in which each content type is supplied) just to be able to view the content it has just received. Content provided as backup by some of the services might not come as user expect, coming for instance in a unexpected format or not following the organisation scheme used in the service (for instance if in the service the content is organised into collections, when it arrives as backup it will come all mixed up in a flat folder). This issue, of not receiving parts of the contextual information, actually pops up in different forms in the previously mentioned backup services from Facebook and Twitter, where it manifests itself for example in Facebook's backup of the friends list which is a simple list with names without any additional information that allows the user to distinguish two persons with the same name. Also transversal too all these tools is that regardless of the well meaning declarations by company CEOs that content belongs to the users, and therefore that they should be able to take it anywhere, they are not openly advertised but instead buried deep within the application settings/preferences dialogue, where a portion of the users are unlikely to find then.

On the other end of the spectrum we have tools that encourage their users to visualise or talk about important (or not so important) events in their lives. Time-line tools like Lim Chee Aun's Life [63] offer users the opportunity to mark and visualise events from their life in a simple time-line format. This type of data visualisation is effective when dealing with time based data, allowing users to browse and assess large amounts of data, identifying trends or patterns that might otherwise go unnoticed. As with all all tools, Life is only as effective as the data that is fed into it. By relying exclusively on users to manually input events, that they might already have marked in other Personal Information Manager (PIM) tools such as calendar tools, Life encourages the adoption of a "pet project" mentality that can be reflected in supplying the tool with incomplete data or even eventual abandon of it by the users due to sheer boredom of the task ahead. Some of events which would be likely to be entered as data for the Life visualisation tool are also very likely to have generated other content types

(for instance photos from attending a given even or from vacations), yet there is no way to establish a relation between that content and the corresponding event. So while a tool like Life can be used to provide an overview of the events that surround or during which certain pieces of content were created (thus contributing to establishing their context), that overview remains disconnected from those pieces of content and can not be used as a search platform or search help by users to rediscover content from those events.

Regarding content discovery, even in digital libraries or repositories, one of the main ways to discover content is by using keyword search, complemented by additional (also querieable) metadata. This kind of search might work well when the underlying content is mainly textual in nature and if the user remembers specific terms that appear in it. Yet the requirement to remember exact words to rediscover a piece of content is a problematic, given the way human memory appears to work. Overly simplifying the subject (for further reference consult for instance [64]), while short term memory is primarily acoustically encoded (i.e. it relies on acoustic cues to store information), which has the side effect of provoking confusion when recalling sequences of similarly sounding words or letters [65], long term memory is primarily semantically (and to a much lesser degree visually) encoded (i.e. it relies on semantic similarity and visual cues to store information), which has the side effect of provoking confusion when attempting to recall semantically related terms [66]. This last effect can manifest itself when users replace one term with another semantically related term and should be taken into account or even exploited when designing search systems, for instance by creating search methods that allow users to tap into their long term memory multiple semantic connects that are recalled when they attempt to describe a given content piece instead of relying only in exact details present in the content itself. An implementation of this approach to search can be seen in [67], which describes a narrative based interface for content retrieval dubbed Quill. Quill is based around the concept of telling to the computer the story that surrounds a given content piece that the user needs to locate. In Quill's interface, the story is presented to the user as a simple text area that must be filled, yet instead of expecting the user to write the entire story from scratch and attempt to interpret it, Quill uses a guided fill-the-blanks approach to story telling. This approach was selected after establishing through a series of interviews, that user produced stories tended to follow common archetypes and were riddled with similar features. To capitalise on their similarity Quill has a common starting point (the author of the content being described) and goes on suggesting additional narrative elements in the order they appeared in the stories produced by users during the test interviews. Narrative elements introduce sentences to the text area with gaps that can be filled by the user to supply additional information. These gaps are filled with the help of specialised dialogues that help to remove the ambiguity from the text, and also give thee user the possibility to state that a certain event didn't happened or that it doesn't remember if it happened. Since the text is always present, the user can go over it to see if it can recall any more information that might help to narrow search results. While the user is concocting a story, the system displays thumbnails (when possible) of the content pieces that match the story told by the user, updating them each time it receives more information. This strategy gives users another opportunity to exploit their long term memories by tapping into residual visual clues that are also used to encode information. Quill is supported by an automated information gathering system that is able to gather information from multiple sources, ranging from the content itself to email, calendar or application activity. Gathered information is integrated into a local RDF based knowledge base that underpins the story driven search mechanism. The use of a knowledge base that is locally stored and only contains data gathered from the local device

is touted as a security advantage, yet relying only on information that can be gathered from the local device hampers Quill now that the user's daily digital activity (and consequently content) is distributed by multiple devices. Furthermore, while Quill lets users search for content by exploring semantic relations (under the guise of story telling), it does not provide any way for users to navigate freely within those same relations. Relations are automatically followed to lead users to matching content, whose discovery is assumed to be the end goal for users and in essence precluding side navigation (such as casual "reminiscing" browsing) that uses the matching content as a starting point.

All of the previous mentioned digital library, digital repository and assorted projects describe parts of what would be needed to construct a digital archive from the bits and pieces of our digital lives. Yet, perhaps the most ambitious experiment done to date in the subject is the MyLifeBits [68] project. While it started as a project that attempted to collect, organise and link digital information, inspired by Vannevar Bush's Memex [69] concept, it has transcended its original objective (and in some aspects even Bush's Memex vision) and become an experiment on the creation of a searchable digital memory archive that integrates both the digital and physical parts of our lives. MyLifeBits software is composed of a database that stores both data and metadata collected from users' daily activities, be them web browsing (where the visited pages are captured), dealing with emails, or just even the user activity (the mouse was moving on this device while a given application was running). Relations between content pieces can be implicitly inferred by the system using time as a reference field (as proposed in the original Memex) or can be explicitly asserted by the user with the creation of labelled links between those content pieces. In addition to individual links, MyLifeBits' organisation scheme also contemplates broad groups of related content, under the guise of hierarchical collections. These are akin to folders in traditional file systems, except that they also explore the linking mechanism by allowing all content (including collections themselves) to belong simultaneously to multiple other collections instead of belonging to a single parent collection. The extensive use of links throughout MyLifeBits makes the system behave like our long term memory, storing semantically (where semantically is here used in a very broad sense) related information and allowing its user to "*follow*" those same relations until it arrives at the desired content piece. In addition to managing born digital content that its users create or gather, MyLifeBits can also be integrated with hardware sensors that record events from the physical world. A specialised wearable camera, dubbed "SenseCam" can automatically take photos and record conversations as its wearer goes around through its daily routine, producing a multimedia digital event log of notable events that can, at the end of the day, be uploaded and stored in the MyLifeBits infrastructure. This log can be further enhanced when it has access to geographical coordinates gathered directly from embedded Global Positioning System (GPS) sensor within the environment capture devices or from standalone trackers, which allows the system to be used in the future to recall not only what its user was doing but also where he was doing it. MyLifeBits project leaser, Gordon Bell has gone a step further by uploading to the system its entire physical archive, scanning everything from old research papers to incoming bills, effectively ensuring that everything that everything he previously produced or received would be stored alongside new born digital items and scanned versions of future physical-only content he received. Despite this effort, there are still some content that cannot be part of his archive, for instance the books he read, mainly due to legal issues. Initial estimates [70] pointed out that about 1 terabyte of storage space would be enough to hold everything an individual would gather from 60 to 80 years, yet this estimate needs to be revised as usage patterns and content evolve. For instance it is now common

for users, particularly those with Digital Video Recorder (DVR) capable set top boxes, to pre-emptively record television programs (i.e. not because he saw them but merely on the assumption that he might want to see them in the future). This usage pattern can throw a wrench in the assumption that MyLifeBits should be a perfect surrogate memory, though it can be solved with the introduction of additional links that distinguish between recording the content and actually see that content in order to avoid the creation of a "*false memory*" (when the user records the program but never actually decides to see it). On the other hand, storing rich multimedia content is a sure way to increase the storage footprint required by MyLifeBits, especially if we take into account that in addition to 3rd party content (such as the previously mentioned television programs), it will also have to deal with home grown content that has become increasingly common with the photo and video captures embedded in every smartphone. Since the initial estimates were made, user generated content has grown both in quantity as well as in complexity (i.e. higher resolution photos and videos, music stored at higher bitrates, larger and more complex documents) which makes it easier for users to accumulate in excess of 1 gigabyte of content per day. The availability of storage space might not even be one of the more pressing issues (at least as long as storage technology continues to improve with the introduction of higher density products at decreased costs) that a project such as MyLifeBits faces. While Gordon Bell had an entire team to help him digitise and organise his past life, the average user will have to deal with that himself. Even if we take into account just the born-digital content, establishing links between content pieces still requires a significant amount of manual labour when said content provides little usable metadata or is gathered without its context. Once again, Gordon Bell had his team to back him up, by designing tools that could semi-automatically discover links and add metadata to the content he had produced, a perk that the average user will not have. Another potential issue with a system like MyLifeBits is privacy, both of its users and of those who happen to cross their path, particularly when the personal log contains content gathered from sensors such as SensCam that can either intentional or accidentally capture unaware strangers. Metadata, a basic requirement to generate (either manually or automatically) the links that make MyLifeBits usable, is now seen from a different perspective. The disclosure of the existence of spying programs such as PRISM [71] exposed the importance of metadata, its potentialities to connecting seemingly unrelated individuals or topics and made it a household word but had the unfortunate side effect of making the general public to shudder at anything that included metadata collection, such as MyLifeBits. This creates a division between those that would be willing to be under surveillance (or more accurately "*sousveilance*" [72] since individual users would be the ones doing the actual recordings and not a centralised entity such as the state) and those who might fear that the resulting records can be used against them (even without the knowledge or consent of those who generated said records). A final issue, that is shared with more traditional repositories is the one of digital obsolescence. Since MyLifeBits stores not only metadata but also the content itself, it must ensure that said content remains in a usable state. Gordon Bell in the book [73] that chronicles his experience while test user of the system has come across with this issue, noting that sometimes some documents produced in an earlier version of a given application would fail to open (or wouldn't be presented correctly). Here MyLifeBits takes a traditional stance recommending to archivists (or their systems users) that documents should be constantly converted to current formats or if not possible, that entire older systems be emulated so that at least their content can be retrieved. At the same time it is also acknowledged that this issue will probably spawn an industry by itself [70], a scenario that once again appears to point out the need for a support team in

order to make this kind of projects viable.

2.3 Users' Behaviour

As important as having the tools to collect, organise and track content is the users' behaviour. At first glance one might say that the average user is not overly concerned with ensuring long term access to his digital objects, or for that matter even aware that they might be in danger. Instead, users appear to be focused on the much more practical goal of being able to retrieve specific digital objects or content nearly from anywhere as long as either they need them or remember them. This leads to the creation of home grown schemes that exploit existing systems that are already familiar to users. Several studies [74, 75, 76, 77, 78] consistently highlights that users send emails to themselves with digital objects that they consider relevant in order to keep them in a known place. This particular solution when coupled with web-based (or as they are now being re-branded "cloud") email provider can be used to create an ad-hoc repository, particularly if the email provider offers strong search features. The same behaviour that leads users to use email as a makeshift repository can also lead them to use specialised services for each content type, so it is not far-fetched to claim that there might be users who exclusively use, for instance, Youtube for personal video management, Flikr for personal photos or Dropbox for remote storage of assorted personal files. The users daily (and increasing) interaction with these kinds of services signals a double shift in the computing paradigm, since they now go as far as perceive them to be linked with mobile devices (such as smartphones and tablets). This in turn has reflexes in the growth previsions for the mobile devices market [79] and in the mobile devices themselves, with Carolina Milanesi, one of the vice presidents of the advisory firm Gartner stating that "[s]oftware and chipset architecture are also impacted by this shift as consumers embrace apps and personal cloud" [80]. The other place where the paradigm shift is felt is in in the way users maintain their digital objects, with an increased reliance on cloud services to keep their content as opposed to keep content on personal local devices, to the point where (if this trend holds) "*[i]n the very near future an archivist might enter the office of a deceased writer and find no electronic files of personal significance*" [81].

While the users' chief priority may be have ubiquitous access (as opposed to continuous long term access) while they remember or need a given content piece, that does not mean that they do not recognise how at risk their content might be. In a survey, when asked about how at risk their content (which for the purposes of the survey meant web based email or blog services) might be, 87.73% of the participants declared that they were aware that their content was at risk [82]. This shows that despite the use of these services (and by extension, other content specific ones) as interim repository alternatives, the majority of the users who do so are actually aware that this type of home-grown schemes in the long term are brittle. According to the same survey, 72.46% attempted to take some kind of mitigation measure such as the creation of backups, mainly to digital supports (local media or replication to different service providers) but also to paper when possible. Ultimately survey participants complained about the lack of backup or archiving tools that don't lose important metadata (such as relations between content pieces) during the backup process. As for the reasons invoked to take those mitigation actions, they range from the rational ones, such as the need to maintain personal records to emotional ones such as the feeling of nostalgia that going though older content can invoke. Nevertheless, both the rational and the emotional reasons have something in common:

for backups to perform as intended they need to contain more than the raw files.

Like their physical counterparts, personal digital objects can carry with them some form of emotional charge. As with their physical counterpart, this emotional baggage is likely to trigger in two scenarios: when their owners interact with them after not seeing them for a while, or (albeit obviously not the same emotional charge) when others have to interact with them when confronted with the loss of the original owner. The first scenario is likely to appear when, either by need or choice, users of a given digital platform have to trim their amassed digital objects. It also both leads to and plays an important role on the question of what digital objects should be kept (or collected). While it might be tempting to dismiss storage space as "cheap" and plentiful (potentially infinite in cloud storage scenarios), and thus stating that every single one should be kept or collected, this is not the case. Storage space is indeed a finite resource (or at the very least the financial resources to support it are) and not all digital digital objects are going to carry the same emotional (or even a positive) weight for their owners, with some being actually a liability in certain situations (such as the one experienced by David Miranda in 2013 [83]). The problem is that there is no sure way to predict which ones will keep a reasonable emotional cargo in the future, making them prime targets to be placed in digital repositories and which one will just become extra luggage. Although it is possible to make educated guesses about what is and isn't relevant, at its core this is not a technological issue and as such the chances of ever create a comprehensive technological solution are reduced. As collections (be them formal or simple accumulations of content) grow, so does this issue. Meanwhile, multiple factors "conspire" to discourage even the most technological savvy users from punning their collections: from the breakneck pace of modern life to the relative low reward to be gained from this task. So if neither automated techniques nor users themselves can decide what should be kept what is the fate of nearly all digital objects? As time goes by they are forgotten (regardless of being "local" objects or remote objects that reside in online services) and enter a state of neglect akin to what happens to physical objects stored somewhere in the attic. When the time comes to replace the device or service where they currently reside, they might be transferred via bulk migration, but expecting anything more than that to happen is unrealistic. In itself this state of neglect is not inherently bad and can be seen as type of natural selection for content (after all it is the way some historical artefacts such as the Dead Sea Scrolls survived), yet it just does not offer any type of warranty that they will remain in a serviceable state in the future (again, some historical artefacts have survived... as rubble and ruins), especially for digital objects that can be rendered obsolete in a single generation (as opposed to the multiple generations required for instance to a language to be considered dead). Nevertheless, since it requires little to no effort (as it happens naturally) it can be considered to be the most common archival/preservation strategy.

The second scenario emerges as part of natural life circle. The death of anyone is always a difficult subject, particularly for those closed to the deceased. If the deceased didn't left specific instructions, it falls upon those who were close to them to be in charge of the funeral proceedings and of taking care of any outstanding issues, all while going through the mourning process, being that even in these proceedings it is already possible to feel the influence of technological advancements [84, 85]. In the near future, a substantial part of the outstanding issues with which relatives will have to deal will include interaction with the digital assets of the departed, be them open accounts in online services or collections of digital objects amassed during life. While interaction with local collections (i.e. those that remain in local devices or storage media) might be feasible without any special provision (assuming that

the storage volumes were not encrypted), dealing with online content and accounts can be more troublesome. Without instructions detailing how to access those accounts (important passwords, answers to security questions, etc) retrieving the content they might contain post mortem might reveal itself to be a near impossible task. Service providers often include provisions in end user licence agreements stating that accounts are not transmissible, which can be used as legal justification to deny access to the next of kin when they attempt to contact the service provider directly, thus potentially locking them out of part of their digital inheritance. Another facet of this issue comes from the rise in popularity of the walled-garden approach (embodied by iTunes or the gaming console marketplaces), where users buy not the content but licenses that allow its use that one can assume that are also not transmissible and expire with their original owner, once again depriving their heirs from part of their digital heirloom (unlike what would happen for instance with a record or film stored on physical media). On the other hand, according to John Troyer, students (age 18 to 23) when asked to imagine a scenario where they have just died were terrified by the prospect of someone being given full, unrestricted access to their digital lives and accounts [86]. While the uncertainty of what would happen to their digital content and a perceived lack of control plays a part in their reactions, one can not dismiss the possibility that those accounts also contain private material that was never (or at least not immediately) intended to be seen by those that now (theoretically) stand in line to inherit it (parents, guardians, next of kin, etc). In such situation, unrestricted (or at least decontextualised) access to what is left behind might not be necessarily a good thing. Meanwhile, enterprising companies have developed services (DeadSocial [87], Everplans [88], Virtual Eternity [89], LivesOn [90] or Eter9 [91]) that explore online material left behind in other services after their subscribers death. Offered services range from transforming social media accounts into memorials for the the departed, to appointing and providing digital executors or even to use an artificial intelligence to continue to run their social media accounts as if were the original owner. Service providers are also not standing still and are beginning to offer their users options within their social services to manage and guide how their content or social profile should be treated in case of extended periods of inactivity (for example Google's inactivity manager [92]). For those users who know about them and took the time to configure them, having the options to manage their "digital afterlives" embedded within their services (as opposed to having to rely on those same options being provided by a 3rd party, that may fail and disappear [93]) increases the trust in that particular service provider. From the service provider's point of view, it provides a legal alternative to keep using information and content that can in the long term be monetised while potentially preventing public relations disasters [94].

Humans as a species are inherently social, a trait that is mentioned throughout history by philosophers from Aristotle [95] to Marx [96] and researchers alike [97]. Being naturally social opens the possibility of taking advantage of group dynamics to solve problems [98] or to promote ourselves within our groups. While direct, deliberate action to ensure access or keep track of a given piece of content or digital object might yield a negligible reward in the eyes of a typical user, sharing the same content or digital object with others can yield a significant reward in the form of "*social capital*". This phenomenon can lead to the digital proliferation (or as Kurr Bollacker called it "*digital promiscuity*" [99]) of certain types of digital objects from person to person. The existence of multiple copies of a digital object spread throughout multiple locations and individuals can be seen as an insurance policy that increases the odds of that a digital object will either be remembered by someone or at least survive the passage of time, forming the underlying concept of the Lots of Copies Keep Stuff

Safe (LOCKSS) [100] program. As new social networks or media sharing services appear, users flow to them searching for another way to gain "social capital" by sharing their digital objects, some of which they might have acquired from past interactions with other services. This results in a net incentive to share, particularly across multiple services (with distinct communities), potentially increasing the odds of creation of multiple copies of the shared digital objects. While one could think that "*digital promiscuity*" could actually contribute for preservation strategies similar to those adopted by LOCKSS there is no evidence that social or media sharing services actually keep multiple copies of the same digital object simply because it was uploaded by different individuals. Adoption of data deduplication technologies that are able to detect duplicate files (or in some cases even duplicate low level disc blocks) under the guise of storage rationality means that it is entirely possible that the multiple copies of a digital object, submitted by different users, be reduced to a single object (be it on live systems or on data backups) [101], at best negating some of the positive effects that "*digital promiscuity*" might have had and at worst increasing the risk of losing the digital object. On the other hand, users also (inadvertently) contribute for content consolidation when they choose to share (or embed) a link to an existing copy of a content piece instead of uploading a new copy. One can argue as that a direct consequence of this approach is that while the probability of finding an individual that actually remembers a particular content piece or digital object increases (since there are now more individuals exposed to it), the odds of a copy of the content to actually survive the passage of time remain largely the same.

Nevertheless social networks and media sharing services should not be completely discounted when it comes to personal digital repositories. Besides the obvious statements that they are popular services, they have become part of the routine of many individuals in which they have invested heavily in order to ensure that the "*social capital*" acquired from them is positive. This quest for positive "*social capital*" brings with it the important feature of motivating users to properly contextualise the digital objects or content that they share. Individuals are willing to dedicate a significant amount of time and effort establishing a proper context for their shared content, in order to ensure that others understand and interpret shared content in such a way that they are seen in a positive way, thus gathering more "*social capital*", which in some social networks can be directly measured (i.e. Facebook's like or Reddit's karma). As such these services have the potential to become sources of both digital objects and of its associated context. While it might be tempting to dismiss the information that can be gathered from such unconventional sources (that here means not only social networks but also other forms of digital communications or activities) as irrelevant, it is in fact everything but that. As William Kilbride, executive director of Digital Preservation Coalition put it, digital objects "*(...) are so ubiquitous that it's easy to treat them as disposable* [102] and nowhere is that more evident than in those that are created exclusively as share vehicles for social networks. Following this logic, social networks can be seen as the digital equivalents to archaeology middens [103], since both are primarily comprised of material that was discarded after it had done its job, yet despite its status, when studied can still provide valuable insights about the daily activities that might be absent from more formal records. One key point that should be taken into consideration when gathering content from these digital middens is that unlike their physical counterparts, that mainly reveal the activities of an entire community, these will mainly reveal the activities of individuals who are actively engaged in a quest for positive "*social capital*". This quest has the side effect of causing a slight misalignment between the online and offline identities of social network users, with the online identities (and with it what is shared and how it characterised) being closer to an idealised form that they

haven't yet managed to implement in the real world [104], and thus some care must be taken when analysing those digital artefacts.

2.4 Chapter Conclusions

This chapter provided an overview of the existing models, approaches and technologies that had the potential to be relevant when designing a personal digital repository system. Additionally, it also provided a bare bones analysis of the expected behaviour of the users of said systems. From the technological overview it is possible to conclude that there are several well-established models for digital repositories that are (either partially or completely) adopted in existing implementations, with particular emphasis on the OAIS model. All of the described repository models assume that there will be multiple, distinct interested parties with well defined roles (often by legal binding agreements) within the repository. The formal agreements and definitions that bind each of the parties dictate their expected behaviour, establish the type of their contributions to repository and can even go as far as to dictate the legal ownership of the content present in the repository. Furthermore, in the described repository models, the repositories themselves exist both to preserve digital objects and to cater the needs of a designated community, which is usually reflected in the model's assumption that a repository's target scope will be very narrow. Ensuring that submitted material is both relevant and within the scope of one of these repositories requires an extensive and thorough content selection process, often performed by specialised personnel, even in scenarios initially described as "personal" such as Paradigm's active politicians pilot scenarios. All of previously mentioned assumptions fall flat in truly personal scenarios. These kind of scenarios replace the community with a single individual that must fill every role defined in traditional models (from content submitter, to curator to end user). Personal interests are fickle, some being long term stable interests, others temporary interests influenced by the repository owner current social-economical context or by current events. Since the content of any personal digital repository is bound to reflect its owner shifting interests, it does not fit the pattern of a highly focused repository that is advocated by existing repository models. Without specialised personnel (in a personal digital repository its user is expected to fill all available roles) to guide the content gathering and selection process, it is highly likely that said process will be chaotic at best and non-existing at worst. Maintaining a full-time cooperation with professional trained curators is usually not a viable option outside specific scenarios (as the Paradigm one) for multiple reasons: from the (possibly sensitive) nature of content to prohibitive costs for the repository owner to the pressure it would place on the professionally trained curators themselves if deployed in large scales. Additionally, the described models place a great emphasis on ensuring or establishing the ownership of the content that is placed into a repository. Repositories in personal scenarios would need to deal with content that albeit produced (or modified) by the repository owner has also been published (or more often than not, shared) in another service, where the simple act of using that service grants the service provider a very broad license over said content, effectively curtailing the ability of its creator to formally establish a canonical version of that content, as often is required by traditional repository models. On the other hand, content ownership in personal scenarios can also be problematic, since the user's digital estate will probably contain content to which the user has a license (as opposed to complete ownership) that as previously stated limits what can be done with it, including placing said content in a digital repository. All of these issues point out that existing models, while effective

in institutional scenarios, are not flexible enough to be extended to personal scenarios without having to leave behind some of their core assumptions.

Existing solutions used to implement digital repositories also fall a bit short when it comes to establish personal digital repositories. Experiments such as the DSpace based personal digital repository [39] and Fedora based Paradigm prototype repositories [45] require users to cope with their institutional repository heritage. As such they either expose a complex and protracted content submission process, that asks individuals for information that might not be relevant (or that they might not even have) about the content they wish to submit, or flat out still rely on experts to perform content selection, an approach that as discussed earlier is hardly scalable. In both cases, asking individuals with at best minimal training to interact with complex software can create an initial barrier that discourages the future use of personal digital repository software. These throwbacks to DSpace and Fedora's roots as software designed to build institutional repositories serve as a warning, that simply throwing re-skinned versions of existing software into a new environment can cause more problems than the ones it set out to resolve.

Content collection remains an open issue for existing repository software, that in one way or another rely on deliberate submissions (either manual submission, or automated submissions that stem from existing agreements). While this allows digital repositories some degree of control over the quality of what is submitted, it has the potential to become a nuisance for users in personal scenarios, particularly when dealing with content that has been previously shared (as it requires individuals to "double share", once in the intended service and once to store the content in the repository). Manual submission processes also implicitly rely on having something to submit, usually a document or file of some kind. With more and more personal content being created and transmitted in (nearly) "*incorporeal*" formats (at least from the typical user's point of view), such as text messages (be them via mobile phones or instant messaging clients) or posts on online communities, manual submission (where users actively create a meta-document with the explicit purpose of being submitted to a repository) becomes too cumbersome to be reliably used with these kinds of contents. In addition to sometimes being in "*incorporeal*" formats and ever increasing, an individual's "*digital footprint*" is also scattered between multiple devices and online services. As such it is highly unlikely that relying on manual submission alone would be a reliable way to ensure content would find its way to a personal digital repository, while at the same time it would be difficult (if not impossible) for single individuals on their own to be able to negotiate with service providers formal agreements to send their content to their respective personal digital repositories. The answer to the content collection conundrum may lie on automatic collection and submission processes that capture content as close to its inception as possible. When supported by metadata extraction services, this approach has the potential to simplify the submission process, freeing individuals from having to fill several pages worth of form fields. This is the type of approach that we see in projects such as [57] or in a lesser degree (due to its highly experimental nature) in MyLifeBits [70]. At some levels adopting a completely automated approach can also be detrimental, since it can be argued that it limits users choices regarding what should and should not be collected, or that it may place users in the awkward position of having to correct erroneous metadata created by automated means. Yet, assuming that it is possible to reach an equilibrium that allows individuals to feel that they are still in control, having access to automated tools is one of the more effective and less intrusive ways to both keep up with the production and gather content for a personal digital repository.

As was previously mentioned, existing approaches place an great emphasis on ownership.

Traditional repositories expect to "own" and manage what from their perspective is the authoritative version of whatever content they are entrusted with and by extension own and completely control the location where said content is stored. Complete control over the the storage location, together with proper policies helps to ensure content integrity, since operations that could affect it are either restricted or monitored and logged. The downside of this approach is that there is a (somewhat) centralised location from which said content can be compromised. Unlike biological memories, that (for now) can only be accessed with some degree of reliability by its biological owner, surrogate external memories can be accessed by anyone and will present themselves exactly in the same way to their owner as to third party entities. While the owner of these external memories can try to make it difficult for other to access them (be it by placing mementos in a locked box, or by requiring a password to access digital files), the fact that access to these surrogate holders can in most cases be compelled, with or without consent from their owners by third party entities (for instance due to a judicial order) can make them a liability depending on its content. This presents a challenge for any type of personal repository or archive, since it needs to balance the possibility of losing access to sensitive content with the possibility of having that very same content seized and misused. Thus while Gordon Bell and other lifeloggers advocate that everything, from digital interactions to physical activities should be captured and preserved [73], there are those such as Viktor Mayer-Schönberger that are concerned about the implications of having a perfect recall tool that ultimately can be wielded by anyone, and end up defending the opposite position that we should keep as little as possible in external memories [105]. Victor Mayer-Schönberger book underlines that the tendency for oversharing, mainly through social networks, can bring problems in the future as a misguided post now can come back later to haunt its creator. This position is also subscribed by others such as Google's Eric Schmidt. As a solution, Victor Mayer-Schönberger proposes that individuals exert a significant amount of self control, avoiding technologies and services that can track them or create external memory artefacts, effectively confining those individuals to a sort of "digital abstinence". Given the prevalence of such technologies and services (it is not a stretch to claim that every piece of modern communication technology, from the web browser to "smart" television sets has the potential to track its users) and the social pressures to conform and use said services, it is unlikely that such a remedy can be adopted in any significant scale. Victor Mayer-Schönberger also defends that forgetting is a natural act that should be mimicked in the digital world. He proposes that content should come with an expiration date after which it would simply vanish, effectively changing the default policy from keep everything to (eventually) forget everything. Such a proposal, though technically feasible (demonstrated in applications such as Snapchat [106]), is probably going to be met with some resistance from historians and archivists since it would mean that they would be losing an important resource in future investigations, though it appears to be gaining traction, especially for sharing "disposable" content [107]. Furthermore, while there is no question that not everything is worth keeping, making everything disappear by default is an incredibly short-sighted policy that might end up creating an even worst case scenario than the one it sets out to prevent. Adding artificial expiration dates to content is a technological solution akin to Digital Rights Management (DRM), which for the vast majority of cases is eventually circumvented, becoming a nuisance only for the legitimate users of the protected content. Ultimately an malicious individual determined to attack another using publicly available "time-protected" content can simply produce an unprotected copy for himself copy by exploiting the so called "analogue loophole" (i.e. pointing an appropriate recording device at the content as it is being displayed). After that, it becomes a matter of time until

the original, alongside its context self-destructs and its creator forgets about it, leaving the malicious individual in a privileged position to unleash a personal attack by spread the same content in a different context.

The issue with this type of time-delayed attack is that for it to work, a malicious individual must recognise the value of the content for future use, an issue that is actually shared with legitimate users. There is no proven way to reliably predict that a given piece of content will become important or hold a particular emotional charge in the future. There is some room for compromise in the middle, if one acknowledges that a purely technological solution isn't feasible, but a composite one might be. While the Internet (or the services built upon it) might not forget, their individual users eventually do. Search patterns change and older content keeps getting "buried" by new entries in search results. And here lies the opportunity to establish a "pre-curation" space, a personal digital repository, that keeps track of some of an individual's "digital footprint", keeping it in (at least temporal) context. A determined malicious individual would still be able to "dig up" potentially harmful content about another individual, yet this individual now has access to an additional tool, his own personal digital repository that might contain material not available elsewhere to help him fill the gaps of what he might have forgotten. Being a "pre-curation" space means that content in it wouldn't need to be held to such high standards of relevance as in traditional digital repositories; in fact its content would only need to be relevant to its owner. Unlike more traditional repositories, it also needs to recognise that the interests of its owner are bound to change throughout time, which is yet another reason for having some leeway regarding what constitutes relevant content. Also unlike a traditional digital repository, emphasis shouldn't be placed on actually owning (as in claiming to have the single authoritative version of a given content piece) but on tracking the location of produced content, in order to sidestep the issue of "double sharing" and blend as seamlessly as possible with its owner already established routines and service usage patterns. Only content inherently at risk, such as "incorporeal" messages that reside in mobile devices, should be copied to the personal digital repository. Such a strategy creates less of a content repository and more of a meta-repository that can be focused in the task of keeping context, effectively making a personal digital repository more akin to a meta-repository, in the sense that the bulk of its "content" would be metadata about the real content, while at the same time it taps into the existing "cloud" services to provide surrogate storage of the real content. Like more traditional repositories, a personal digital repository should offer its users the tools to logically organised their content in collections, being that said organisation would necessarily different from individual to individual. None of projects or approaches presented in this chapter is able to provide these functionalities to their users in a coherent package, and as such there is the need to create a new breed of digital repository specifically tailored to deal with personal scenarios.

Chapter 3

Repository Architecture

"It is desirable every thing printed should be preserved, for we cannot now tell how useful it may become two centuries hence."

Christopher Baldwin, January 10, 1834

Proceedings of the American Antiquary Society, 1812-1849

As seen in the previous chapters, there isn't a solution that allows individuals to keep track of the content produced throughout their daily routines. While some of it might at first appear to be nigh irrelevant, that is a label that can only be applied with the benefit of hindsight and probably not even then, as it is notoriously difficult for individuals to accurately gauge how valuable their produced content will be if it ever reaches an historian 100 years into the future, or simply their next of kin in the not so distant future. Since it isn't possible to assign to every single person on the planet an archivist with the explicit tasks of gathering, categorising and preserving the digital content that their assignees produce, the next best solution is to delegate that task to the individuals themselves, with all the inherent pitfalls that such delegation entails. Those who choose to take upon themselves the task of keeping track of their own digital content need to be supported by novel tools and approaches in order to ensure some degree of success in their endeavour. The new tools and strategies required for such a scenario need to strike a balance between effectiveness, intrusiveness and perceived usefulness: they need to be effective enough to collect content that would otherwise be unreachable or difficult to link to a particular individual; they need to avoid being so intrusive that their use significantly affects established routines and their user's need to feel that they can have some sort of immediate benefit from their use, otherwise there is a low probability of adoption.

3.1 An Intertwined Ecosystem

The rise of digital technologies, in particular web technologies that let users produce their own content, the personal content cycle (illustrated in Figure 3.1) has become more prominent. In this cycle individuals, gather information from various sources (including their own past/present point of views and collections) in order to create their own opinions and with

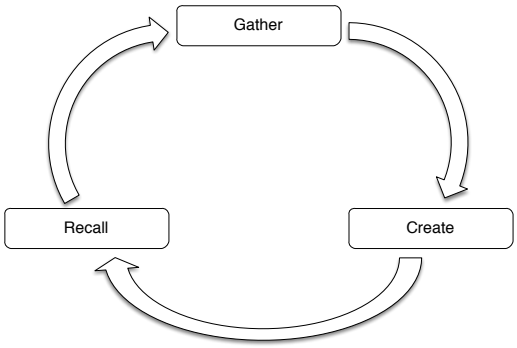


Figure 3.1: Personal Content Cycle

them, new personal content that might be shared (either publicly or privately). When shared, this personally created content becomes part of the collections of others that found it interesting, or if it remains private it becomes part of its producer's collection of content. Content collections remain in standby, with their content ready to be recalled in order to begin the cycle anew. The key issue here is that personal content can be forcefully removed from the cycle when either its creator or those with whom the content was shared forget about its existence, or when although someone knows that it existed in the first place is unable to locate it. Thus in addition (or perhaps more important than) to gather personal content, there is the need to gather information that allows said content to be found again when needed.

Given the possible breadth of topics, types and origins that personal digital content might cover, there isn't a single content collection strategy that would work in all the possible circumstances that can appear in personal scenarios. With the traditional approach of having specialised personnel gathering and submitting content to a curated repository logistically out of question, there is a strong case for the creation of tools tailored to gather content from particular origins or even squarely aimed at gather a certain type of content. While at first glance having a fragmented tool kit might appear to be highly intrusive, and as such the exact opposite of what would be recommended for a personal scenario, the existence of separated tools for each content type or origin also allows those same tools to provide "extra" (from an archival perspective) services that serve as incentive for individuals to adopt them. Furthermore, by providing multiple tools, users can gradually adopt them as they need them (ostensibly due to their other functionalities) as opposed to use a single tool where the majority of available functionalities feel irrelevant for them. A simple example of this approach is a generic file synchronization tool. Under the guise of ensuring that a given set of files can be accessed from every computing device under the control of a given individual, those files can be uploaded to the individual's personal digital repository for safekeeping and analysis. Other tools should follow the same approach and offer a convenient service that encourages their adoption by individuals in order to ensure that content can be gathered and placed in their personal digital repositories. This approach can be particularly important for personally relevant content that from a regular user's point of view doesn't have a direct representation in the form of traditional files, such as chat or text messages, browser history or even social network posts. Such tools are in a position to collect content itself, yet unless it is in immediate danger, i.e. it on a device that can be lost (as opposed to already residing on a "cloud" service) or its format doesn't have a traditional file representation, one can argue that

the most important information that this type of tools can gather is in fact meta-information about the content, particularly the location of the said content. The rational behind this is that a personal digital repository should allow users to locate and go back to their content, preferably in their original context, and only if not possible in the context of the personal digital repository.

Content collection tools are only part of what is needed in order to give individuals a way to keep track of their digital content. Collected content still needs to be organised, and that is where a personal digital repository enters. Unlike traditional approaches, where digital repositories are tasked with content storage, management and dissemination, personal digital repositories are just concerned with content organisation and supporting the collection tools. Though a personal digital repository still needs to have access to services and storage space in order to deal with content that the collection tools deemed to be in danger, its operational focus is in the metadata stream received from the different content collection tools. Metadata collected from different providers can be melded into a shared context, establishing links (that might not be obvious at first glance) between pieces of content, that can be explored by the repository owner in the future to rediscover pieces of content that he might have forgot about. Another issue is that, as was previously noted, humans as a species are inherently social, which implies that some individuals or groups have a shared social context, that might be relevant to frame specific pieces of content. Though generally thought as shared, the social context is in fact perceived in a slightly different manner by each individual, based on previous experiences and current interests. Thus from the point of view of a personal digital repository, the social context can become one more metadata stream gathered by collection agents, albeit one that in addition to be melded with the personal digital repository's shared object context, can also be used to establish links between different individuals (as perceived by the repository owner) and ultimately between their personal digital repositories.

The previously described version of the personal content cycle holds true for both physical and digital content, at least until the sharing part. Physical content by nature can't be easily shared, unlike their digital counterparts, which is one of the reasons why the personal content cycle became more prominent with the advent of digital content. On the flip side physical content, with its requirement for physical storage space is more prone to be "accidentally" rediscovered (for instance out of curiosity to see what is in a box tucked under a bed) than digital content, especially if one takes into account the rise of third party services that host said digital content, nearly removing "storage space reclamation" as a reason to rummage though old folders. This diminishes the chances that digital content has to be rediscovered, triggering some reminiscing or even the possibility of a being given a new interpretation and thus to return to the personal content cycle. It should be noted that from its owner's point of view, content rediscovery can only be considered as a positive outcome, since reminiscing about a given content piece can offer in itself a feeling of instant gratification while new interpretations can lead to production of new content and the renewal of the cycle. The introduction of dual purpose collection tools backed by a personal digital repository aims to foster a change in the personal content cycle by introducing a deliberate (but arguably initially hidden) curation stage as seen in Figure 3.2. The personal digital repository gives its owner a known point where they can go and "take a peek" (akin to looking under the proverbial bed where physical content might by safely tucked in a box) at some of the content they gathered throughout time. The content collection tools serve the dual role of ensuring that content and its associated metadata reaches the repository with minimal interference in the repository owner's routines, while at the same time keeping individuals engaged by providing services

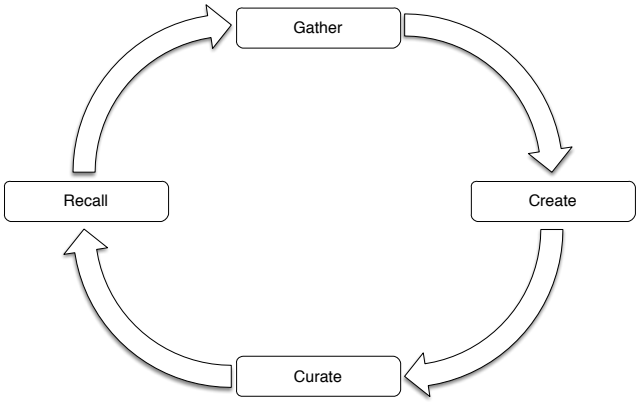


Figure 3.2: Modified Personal Content Cycle

in which they might be interested, effectively creating an intertwined ecosystem for content collection and lightweight curation. Though this approach is unlikely to lead to the creation of collections on par with the quality of those created by trained professionals, it will at the very least provide some insight of how a given individual lived and what he valued, both for the individual himself, or their heirs in the future. Furthermore, should a memory institution ever be granted access to a personal digital repository, it could act based on the content contained in the personal digital repository, by enacting additional preservation measures or using the contained materiel in future historical reassessment studies.

3.1.1 Content Collection Guidelines

In this scenario the adoption of dual purpose tools meant to be used by (nearly) untrained individuals rather than highly skilled professionals is in itself a notable departure from existing approaches, yet it is not the only departure proposed. The rise of social media and cloud platforms signals that individuals are willing to trust their content to third parties, often trading full content ownership for the convenience of being able to access it from anywhere, being that there are three major risk factors associated with this behaviour. The first one is that content might eventually disappear from the service where it resided; the second one is that content becomes unreadable due to obsolescence of its container format and the third one is that its owner simply forgets about the existence of that content. The collection strategies adopted by individual content collection tools can help to mitigate (though never fully resolve) the first and third risks, while the repository itself can help to mitigate the second one though it should be noted that it is neither its primary goal or secondary goal (and in fact, it might well be preferable to do nothing than to risk a potential information loss over ill planned conversions).

The risk faced by content in any kind of digital preservation scenarios, particularly in traditional repositories, is relatively well understood being the object of a number of previous studies and models [108, 109] . It is usually dealt with a combination of collection, distributed replication and repository wide policies, to ensure some degree of survivability in case of disaster (be it accidental or man-made). However, in the traditional approach the repository will also try to become the authoritative source of said content. In personal scenarios, one might adopt a similar strategy, gathering content early and often with the help of the collection

tools, but foregoing to assume the role of authoritative source in favour of becoming a last resort replica. Additional replicas can be created, if the underlying personal digital repository has access to the services and resources required to establish them. Besides minimising the risk of losing access to content due to it becoming unavailable from the original source, collecting early and often in personal scenarios has the advantage of allowing the repository owner to see how said content evolved over time, browsing through the iterations that lead to the final content, being that this is applicable to work in progress content such as documents produced over time. Nevertheless it is unwise to discount the contribution that social media and cloud platforms can provide for digital preservation, as it is in these services' best interests to ensure access to their user contents in order to maintain their reputation. Content shepherded by these services can be considered to be in a somewhat more sheltered environment than content that exists solely on the devices controlled by the personal digital repository owner, and thus end up being at a slightly lower immediate risk, so placing a copy of this content into the personal digital repository may not be an immediate concern. Other types of content, particularly non-traditional content such as browser history or messages of various kinds, which by their nature as either device, service or application bound present a greater immediate concern. As such, non-traditional device bound content is the type of content for which a personal digital repository should strive to be the primary authoritative source, and when possible should manage it directly.

On the other hand, although the risk of of a given piece of content to be nearly forgotten is always present in any type of repository (digital or otherwise), it is significantly higher in personal scenarios. Such risk is exacerbated by the high rate of content accumulation made possible by social media and cloud platforms that drive the modern content cycle, and that in some cases imbue their users with a mentality of create, access or share now, think about it later. The net effect of this mindset is the devaluation of the importance of previously personally created (or procured) content in favour of new content that can be re-shared immediately. While it is arguably impossible for a personal digital repository to prevent its owner of forgetting about its own content, with appropriate metadata it can serve as an entry point for exploration of previously collected content. Metadata gathered for a personal digital repository must then be useful not only to characterise the collected content itself, but also to allow its placement in the broader context of the repository. Personal content in a personal digital repository, by its own nature is bound to be more strongly interconnected with other pieces of personal content than content that just happen to thematically fit within traditional repositories. Additionally, as mentioned previously, some of the content that will end up being in a personal digital repository will be available from social media and cloud platforms. If this is taken into account, one must recognise that content in this situation has actually two contexts: one is the personal context in the repository (i.e. how it relates to other pieces of content in the repository) and the other is the broader, public facing context that comes from the platform from where it was originally gathered. While the personal context serves as guide to allow content rediscovery, the repository owner might be interested in being able to go back to the original platform and see how that context has evolved.

Taking these issues into account, collection tools should follow these generic guidelines in order to decide what to collect and when to collect:

1. A collection tool is responsible for defining its own content targets, be them traditional (e.g. files) or non traditional (e.g. messages)

2. Immediately collect content that is application or device bound

3. Prefer metadata collection when the content is available from social media or cloud platforms and can be retrieved later

4. Regarding metadata, in addition to the one that describes the content itself also collect any information that can enrich its context (i.e. original location, original device, temporal data, approximate location of user if available)

3.2 Proposed Architecture

As previously stated section, personal digital content might come from a variety of sources, be represented in different formats and cover a wide range of subjects. This is a scenario that traditional digital repositories avoid, as although their underlying platforms are usually content-agnostic (albeit slightly skewed to favour various flavours of textual content) once deployed they become effectively restricted by political decisions to accept only content deemed pertinent to the community they serve. Content restrictions are enforced through a combination of repository-wide policies (for instance, rejection of content that is not represented in a set of predetermined formats) and a human guided content selection process often performed by experts that act as the repository's gatekeepers. More often than not, the intent of format based restrictions is to ensure that content present in traditional repositories is at least in a well-known and widely used format (inside the community that the repository serves). This serves the dual purpose of ensuring that the vast majority (if not all) the members of the community have access to the content from the repository as well as an implicit safeguard against format obsolescence, since presumably well known widely used formats will have more direct migration paths to other formats should the need arises. On the other hand the role of the human guided content selection process, often performed by trained personnel, is to ensure that the repository content is relevant to the community it serves.

In personal scenarios the responsibility of selecting content to be placed onto the personal digital shifts from trained personnel enacting well defined policies to individuals. This shift makes it difficult (not to say nigh impossible) to accurately predict the nature of the content that will end up in a personal digital repository, thus a personal digital repository will end up covering a wide breadth of topics, each of them with multiple possible content representation formats. What can be predicted is that different individuals will have different levels of engagement regarding their accrued content. For instance, if the exact same photography was to be submitted to the personal digital repository of two different individuals, one could be interested in what that photography depicts (and will remember it for that) while the other might be also be interested in the technical details of how the photography was taken (exposure, lens aperture, or other settings that produce distinctive effects), and use those details to recall that photography in the future. This simple scenario illustrates one of the multiple fractures that may appear in the way that individuals see and eventually come to remember their content, pointing out that a personal digital repository will need to be able to adapt itself to the needs and interests of its owner and not the other way around, even if that means having to introduce different levels of detail for the same content.

Additionally, as mentioned in the previous chapter, studies indicate that human long term memory is semantically encoded [66], a trait that has the effect of provoking confusion between semantically related terms when attempting to recall information. In the context of a personal digital repository this effect is likely to be further amplified by two related factors: the first one is that over time the interests of any given individual tend to evolve, with engagement

levels fluctuating accordingly, leading to the need of reorganising gathered content to adapt to a new reality; the second one comes from the continuous learning process inherent to the human condition that brings with it new concepts and terms that are either semantically related or newly applicable to some of those previously used to describe or organise existing content in the repository, and that individuals might attempt to use in order to retrieve content from their personal digital repositories. These issues point out that for a personal digital repository, adaptation must go beyond the presentation level (i.e. how much detail is initially directly exposed to the repository owner) and bleed into the organisation schema itself. Though it might be tempting to think that the ideal scenario would be to provide with each user its own completely personalised organisation schema right from the onset, such approach falls prey to the same logistic issues that prevents the assignment of a trained archivist to each individual. Furthermore, the adoption of a strategy that does not ensure the existence of a common organisation core would greatly hinder both interoperability between the personal digital repository and its associated set of content collection tools as well as interoperability between repositories in a foreseeable future. Instead, the organisation schema adopted by a personal digital repository will need to take into account and provide the means to encode within itself custom organisation features provided by the individuals themselves. There are also two additional requirements for the organisation schema that underpins a personal digital repository: the first one is that it must be capable of representing the temporal dimension of the personal digital repository; the second one is that it needs to be flexible enough in order to accommodate classification needs that might not be present when the schema was first selected.

Given the previous assessments, the challenge for personal digital repositories is twofold: its architecture must be flexible enough to deal with different types of content that come from multiple sources (albeit all of them acting in the behalf of the repository owner), while its underlying organisation schema also needs to be flexible enough in order to be able to integrate the organisation needs of new content types or to re-frame previously acquired content in the light of newly existing data.

A possible solution for these challenges would be to design a personal digital repository around an extensible modular architecture, whose underlying organisation schema is encoded using semantic web technologies (therefore making it also modular and extensible). In this kind of architecture (seen in Figure 3.3), a personal digital repository is composed by two module types: a core module and an arbitrary number of auxiliary, user manageable service and content modules. Modules of the second type are responsible for the implementation of content dependent or variable functionalities. These may range from providing internal services to other repository modules, such as additional storage backends (both traditional storage space for content proper as well as semantic storage to support the repository shared context), to providing support within the repository for content gathering tools or even to expose additional user facing services, such as content-specific views or enhanced search capabilities. Meanwhile, the core module is responsible for the implementation of user authorisation and authentication services, module management and last resort content management. Additionally, this is the module that defines the common interfaces that other modules use to communicate with each other and with the outside world. Common external interfaces are managed by the core module and serve as proxies, determining to which (if any) a given request should be sent, which implies that the core module must also provide a content registry service.

While one can argue that some of existing repository software already possess (in broad terms) modular architectures, one of the key differences present in this proposal is that the

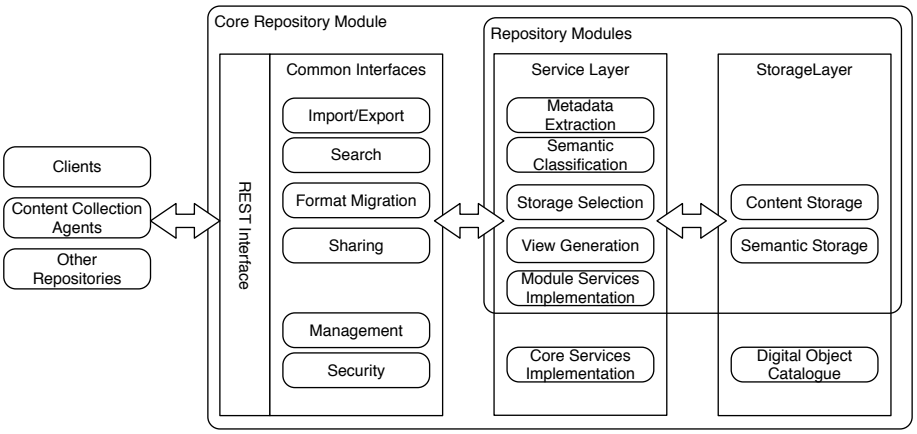


Figure 3.3: Proposed Architecture

repository owner is able to add modules that extend the repository in order to add new functionalities (or to perform a thematic shift) after deployment without the need of intervention by specialised personnel. Despite this trait, "regular" repository owners are not expected to be able to create their own modules, but instead to be able to obtain them from a distribution site (like a module marketplace). This creates an opportunity for those with relevant skills and interests to form a community around the development of said modules and (if any) associated collection tools.

The dependency of the personal digital repository on its collection tools also entails the need to be always connected. As such it makes sense to make the personal digital repository a web application, as opposed to making it a traditional standalone application. Such an approach can further explore the familiarity of users with web technologies by having the presentation layer of the repository (should it need to have one) to be created with existing web technologies, such as HTML, Cascading Style Sheets (CSS) and Javascript, and accessed through a standard browser. It also opens the possibility of eschewing the development of a dedicated communication protocol in favour of using the standard HyperText Transfer Protocol (HTTP) and REST-like endpoints for the content collection tools.

3.2.1 Core Repository Module

The Core Repository Module serves as the seed around which a Personal Digital Repository will be created. It provides basic functionalities and services that allow modules to communicate with each other, which necessarily means that it also contains the specification of the interfaces (both internal as well as external) that other modules must use. The characterisation of the core repository module can be divided in several steps: the first is to define the basic functionalities that a personal digital repository should provide, identifying those that can be delegated to other modules, and those that should be implemented by the core module itself. The second step is to define a minimum set of interfaces that support the previously identified functionalities (be them external or internal interfaces). The third step is to define the minimum set of information that is common to all content (be it intrinsic or synthetic repository-only information), which can be used to store and produce a "base" digital object to be stored within the Personal Digital Repository.

The basic functionalities expected from a Personal Digital Repository are a reflection of the responsibilities of the repository itself. As such, one can reasonably expect that a Personal Digital Repository to be able to store and retrieve the content its associated collection tools have gathered, along with important metadata that describes said content, as it also reasonable to expect that a Personal Digital Repository will allow gathered content to be accessed and managed. Content placed within a Personal Digital Repository should also be kept outside the reach of unauthorised prying eyes, which means that one should expect that ensuring at least some measure of security to be a responsibility of Personal Digital Repository itself. Lastly, given its modular architecture it is also legitimate to expect to have some way to manage and control the repository modules. From this we can gather that a Personal Digital Repository must provide functionalities that deal with content storage, retrieval and management as well as with security and module management. It should be noted that unlike traditional digital repositories, given the potentially sensitive nature of their stores, broad content dissemination is neither a responsibility nor an objective for Personal Digital Repositories, though ensure that the repository owner can access content that is completely controlled by the repository is of paramount importance.

The content storage functionality is an example of a functionality that albeit being critical for any type of repository, in this context can also be delegated to one of the repository modules. While the core repository module can provide a default storage option using the local file system, it is clear that no two personal scenarios are alike and as such there will always be scenarios where the default would be limiting the potential of the repository. Allowing the delegation of storage functionalities it becomes possible to support multiple storage scenarios, from the default scenario to use the resources from the repository's host machine, passing through (local) network storage resources accessed through multiple protocols, to scenarios where the repository owner has access to one (or preferably more) cloud storage services. The core repository module is responsible for the management of the available storage options provided by the various repository modules, marking them as being either primary (i.e. those where content and support metadata should be preferably stored and from which should be accessed) or backup (i.e. those where content should be placed for safekeeping purposes only). The added flexibility that the introduction of external storage options brings comes with the caveat that any storage option that is not fully under the control of the repository owner might fail due to external factors, such as missing payments or even service shut down. As such it is recommended, particularly when using cloud services the simultaneous use of multiple services from different providers with at least one designated as backup. As mentioned previously, for a personal digital repository, metadata may be as important as the content itself and it also needs to be stored. A personal digital repository needs to maintain two types of metadata storage: the first type is an digital object catalogue, that stores administrative metadata [110] pertaining each individual content piece in an atomic package (i.e. something that can be stored and manipulated as whole), as well any support data that the personal digital repository might require (for instance login information, content operation logs); the second type is an interconnected storage used to build and maintain the repository shared context. For the first metadata storage there aren't any inherent technological bias, so it can be based on any type of data storage technology currently available (be it relational databases or nosql alternatives) and use any type of underlying replication architecture. The restriction that exists is that since this metadata storage type will also house the repository's support data it must be both unique and chosen before deploying the personal digital repository. Thus whatever is chosen can be seen as a fixed component for the future layout of repository itself. On the

other hand, the second metadata storage type has an inherent bias due to the nature of its intended functions and the metadata that will keep. As it will be used to keep the shared context of the personal digital repository, whose relations for all intended purposes dictate the organisation and classification of the collected content, there is a strong incentive for this to use a type of semantic storage. Given that it should be possible to reconstruct the basic lattice of relations from the raw metadata content stored in the digital object catalogue, it stands that this functionality can also be delegated to a repository module, albeit much like the content storage functionality the core repository module should provide a default storage implementation. Much like the content storage functionality, the advantage of allowing the delegation of semantic storage to a repository module is flexibility. Semantic storage can be based on anything from an embedded graph database to a dedicated semantic storage server. Additionally different storage options are expected to bring with them different reasoners, that based on the available semantic information can be used to establish more links between content.

In the same way that content needs to be stored, it also needs to be retrieved in order to be useful. In this regard the core repository module's prime concern is to be able to locate content. View generation can be delegated to specialised modules according to content type or the client requesting it. For instance a content collection tool might want a particular format (often akin to the one it used to deposit the content in the repository in the first place) in order to restore content to a mobile device, while the repository owner will surely want a visual representation that allows him to further navigate and explore his repository.

Regarding security, a repository (be it personal or of any other kind) needs to be able to ensure that only those properly identified can perform tasks or access the content it shepherds. This is a critical functionality that has a direct dependency on the availability of a place to store the personal digital repository's support data. That direct dependency, when coupled with the potentially sensitive nature of the content held by a personal digital repository makes it unwise to delegate any security function to a repository module (as the risk for abuse is non negligible) thus security is the sole responsibility of the core repository module. It should be noted that in this context any type of security measures and policies are aimed at external threats. At this time the decision of how to deal with trojan like threats, such as those posed by a malicious repository modules installed by the repository owner himself, is left to the actual personal digital repository implementation. The core repository module is responsible for the implementation of all authentication, authorization and accounting services. At the very least the repository owner must supply a username and password before performing any actions in the personal digital repository. On the other hand, the repository needs to expose endpoints so that the content collection tools can gather and retrieve content, being that not all repositories will need to authorise the same set of content collection tools, with tools being added (or removed) as time progresses. This precludes the pre-authorisation of all tools when the repository is created, yet if one considers these tools as clients that want to access a service on behalf of the repository owner them the repository can adopt the OAuth protocol [111] (i.e. becoming an OAuth provider) to deal with these interactions. One of the requirements for OAuth is that clients, in our case the content collection tools need to obtain client identifiers, a step typically being done out of band with a manual request by the client developer issued to the service developer. Given that each personal digital repository is expected to be independent, it is infeasible for client developers to manually issue such requests to each and every personal digital repository. Instead the personal digital repository needs to provide a semi-automated (to give the repository owner veto power over unwanted

requests) client registration service. As a security related service, its implementation falls under the responsibility of the core repository module. While there is no established protocol for automated or semi-automated client registration in OAuth, there is a proposed version for a standard [112] that can be used as a guide for the required flow. After client registration, a client still needs to identify itself and regularly request authorisation tokens in order to access services. It should be noted that OAuth was initially geared for interactions between web applications, an as such makes heavy use of callback URLs. Collection tools that are unable to provide or access a callback URL can use an alternative out-of-band flow that requires an web aware user agent such as an external browser to obtain an authorisation code on their behalf.

The last major responsibility of the core repository module is to manage all other modules. Additional repository modules provide different levels of functionality, ranging from low level repository functionalities (such as the previously mentioned storage modules) passing through the support for collection tools all the way to interface modules that can provide additional visualisations of the available content and associated metadata. Despite the different functionalities provided, it is possible to identify some properties that are transversal to all repository modules:

identifier An identifier by which the module will be known internally in the personal digital repository. This identifier must by unique among the all those present in the repository.

title A user friendly short title by which the plugin will be known by the repository owner.

version A version number used to distinguish between different versions of the same module.

creator A string identifying the individual or organisation that created the module.

creator email A string that represents an email address that can be used to contact the module's creator.

description A free form description used to describe what the module does to the repository owner.

dependencies A list of other modules (including version ranges) that the current module depends on to work properly.

state The current state of a the repository module used to describe if it is in active use, or simply installed but not loaded.

It is also possible to identify some properties that, while not common to all repository modules or defined by the core repository module, can become important in the interaction between the repository module and the core repository module:

content process capabilities For modules that support content processing, it defines a list of content types that the modules understands and from which it can extract information.

content process priority For modules that support content processing, it maps the priority assigned to the module to process content types from those it claims to be able to understand.

Relative URL	Role
/dataManagement	Content submission, update and removal
/content	Access to repository content
/search	Search the repository content
/auth	Authorization and authentication
/pmh	access to content metadata through OAI-PMH protocol

Table 3.1: Interface URL groups

content transformation capabilities For modules that support content transformation, it maps valid input formats to valid output formats.

content view generation For modules that provide additional views, it defines a list of content types for which the module can generate a user facing representation.

storage level For storage options modules, defines if a module should be treated as a primary storage provider or as a backup location.

It should be noted that the way these properties are represented is implementation dependent. Furthermore, modules should be granted some modicum of autonomy, represented by allowing them to have module-specific configuration properties that can be changed independently from the common properties. Information regarding available modules and their respective state is gathered and kept by the core repository module in the support data part of the digital object catalog. The declared content process capabilities coupled with its priority, is used to decide which, if any, of the available modules will be called upon to deal with an incoming request.

Common Interfaces

The core repository module presents to the outside a set of common interfaces. These represent major operations that can be performed within the scope of the personal digital repository that are roughly aligned with the core module's expected responsibilities, such as content (or metadata) import, search or security operations. Operations are requested using a REST-like (instead of a RESTFul) [113] interface with related operations grouped under a specific URL. The REST-like moniker is applied for instance, as for readability purposes, the name of the operation can be included as part of the URL. For example, under this scheme, a POST request intended to submit content to the personal digital repository should not be made to the "/dataManagement" URL but instead to "/dataManagement/storeContent". It should be noted that repository modules are allowed to expose services that are not covered by the common interfaces, as long a they do it in their own group. Repository modules are also allowed to use and define parameters for operations in addition to those defined by the common interfaces.

Requests sent to any of the exposed common interfaces pass by the core repository module first, that act as a proxy whose task is to determine to which, if any, module the request should be sent. Malformed requests, such as those that attempt to invoke an operation using the wrong verb or with missing mandatory parameters can be immediately rejected by the core repository module. If no module is registered to deal with sent content, the core repository module can act as a content importer of last resort, by attempting to extract as

Operation	Method	Mandatory Parameters	Optional Parameters	Description
storeContent	POST	content or externalContentLocator	contentHandler	Stores content in the repository
	PUT	id, content or externalContentLocator	contentHandler	Updates content in the repository
removeContent	DELETE	id	contentHandler	Removes content from the repository
exportContent	GET	id, format	contentHandler	Exports content from the repository in the requested format
	POST	id, format	contentHandler	Exports content from the repository in the requested format and stores a copy of the output in the repository
inheritContent	POST	content, metadata		Imports inherited content into the personal digital repository
createCollection	POST	cName	cDescription, contentHandler	Creates a collection in the repository
removeCollection	DELETE	cId	contentHandler	Removes a collection from the repository
addToCollection	PUT	id, cId	contentHandler	Adds the specified content to the collection
removeFromCollection	DELETE	id, cId	contentHander	Removes the specified content from the collection

Table 3.2: Summary of data management operations

much information as it can from both the request and the associated content in order to create a basic "black box" digital object. To facilitate the task of determining to which module a given request should be sent, most operations can be invoked with an optional parameter "*contentHandler*" that serves as a hint about which module the caller is expecting to be used to process the request. The use of any service, be it exposed through the common interface or module specific is conditioned to acquiring proper authorisation, a process that should be mostly handled by the OAuth protocol (as previously described).

The "*/dataManagement*" URL groups together operations that deal with content submission, update and removal. The group also provides operations to manage collections (i.e. arbitrary groups of objects explicitly created by the repository owner) and to receive inherited content. This group provides the proxy for the following operations, whose summary can be found in Table 3.2:

Operations over content

- "*storeContent*"
- "*removeContent*"
- "*exportContent*"
- "*inheritContent*"

Operations over collections

- "*createCollection*"
- "*removeCollection*"
- "*addToCollection*"
- "*removeFromCollection*"

On the content management side, the "*storeContent*" operation is responsible for handling content submission. When invoked using the POST method it will add new content to the personal digital repository, while when invoked with the PUT method it will try to update previously submitted content. Content can be represented by the collected digital object itself, or by a handle that allows it to be retrieved at a later stage. This scheme serves to support the collection guideline of giving priority to metadata collection when the content is available from social media or cloud platforms and can be retrieved at a later time. Content itself can be supplied in the request body in the "*content*" field, or if needed as a multipart attachment, provided that it contains at least one part named "*content*", while the content locator can

be supplied using the "*externalContentLocator*" parameter. At least one of these parameters must be supplied in order for the operation to be successfully completed. When invoked with the PUT method, the content identifier should also be supplied using the "*id*" field along with either a new version of the content, or updated metadata. The expected result of this operation is, if successful the placement (or update if invoked with PUT) of one or more digital objects created from the supplied content in the personal digital repository.

The "*removeContent*" operation is responsible for content removal and must be invoked using the DELETE method. This operation has a mandatory parameter, the identifier of the content it will attempt to remove from the repository, which should be supplied using the "*id*" field. While in principle it should go untouched, this operation provides a means for owners to excise from their personal digital repository content that they was either accidentally collected or that they deem unfit to be there, since the personal digital repository represents its owner's interests and vision of the world (with all the bias that can come with it). How to remove the digital object (directly or first to a removed content area) is left to the implementation, as long as the expected result of removing the digital object from the "main" repository representation is achieved.

The "*exportContent*" operation is responsible for handling content transformation to a different format. Thought as a format migration tool helper, it requires two parameters, one being the identifier of the content to be exported, supplied using the "*id*" field and the format to which the content should be exported, supplied using the "*format*" field. The core repository module does not impose any restrictions to the structure of the "*format*" parameter leaving its interpretation to content handling modules, though for optimal results the requested format should match one of the registered formats. In case the core repository module can't determine which module should handle the request, it may attempt to serve as an export of last resort if it understands both the requested content format as well as the requested output format, though this behaviour is optional and implementation dependent. This operation can be invoked with both the GET and POST methods. When invoked with the GET method it will retrieve the requested content in the requested format (assuming that it is possible to export the content to the requested format) while when invoked with the POST method it will export the requested content to the requested format and then store it in the repository for future use, with the restriction that the new content metadata must define a relationship that links it back to the original source.

The "*inheritContent*" operation is responsible for handling content inheritance. It is invoked with the POST method and has two mandatory parameters, "*content*" and "*metadata*", that represent respectively the inherited content itself and the metadata associated with it in the previous holder's personal digital repository. This operation differs from the others in that it isn't possible to use a "*contentHandler*" parameter to control who should deal with the incoming content. Instead this task can be performed either by the core repository module or by a repository module, provided that only one is registered to implement this service. The adoption of such a strategy allows the introduction of new modules that can eventually deal with incoming content from other implementations of a personal digital repository. Furthermore, due to the specific nature of this operation it is highly recommended that it requires a special security permission to access.

The "*createCollection*" operation is used to create a new collection in the personal digital repository. A collection serves as a user defined arbitrary group of content used as an additional organisation tool. There is no restrictions to the number of collections to which a given content piece can belong to at any given time. New collections are created by invoking the operation

using the POST method, which has a mandatory parameter "*cName*", that represents the user friendly name by which the collection will be known in the repository. Although an internal identifier will be assigned to the collection, to avoid confusions collection names should be unique. There is an optional parameter, "*cDescription*" that can be used to provide a human-readable description of the collection (for instance its intended purpose). Specific collection types can be created by hinting which module should handle the request with the "*contentHandler*" optional parameter.

The "*removeCollection*" must be invoked with the DELETE method and is used to remove a collection from the repository. It has a mandatory parameter "*cId*" which represents identifier of the collection to be removed. It should be noted that this operation should only remove the collection (if it does it directly or first to a removed content area is left to the implementation) and links to the collection content. As a rule of thumb, removing a collection should not remove any of the associated content from the repository.

Content management within a collection is done through the "*addToCollection*" and "*removeFromCollection*" operations. These operations have two mandatory parameters, "*cid*" that represents the identifier of the collection and "*id*" that represents the content identifier. Invoking the "*addToCollection*" operation with the PUT method will associate content with a collection while invoking the "*removeFromCollection*" operation with the DELETE method will sever an established association.

The "*/content*" URL groups together operations that deal with content or metadata access and retrieval. Operations in this group are not allowed to affect the state of the repository, a fact that is reflected by restricting them to be invoked with the GET method. Though created through different operations (or even implicitly), for the purposes of these operations, collections are just another piece of content that resides in the personal digital repository. The group provides a proxy for the following operations, whose summary can be found in Table 3.3:

- "*listContent*"
- "*showContent*"
- "*retrieveContent*"
- "*showMetadata*"
- "*retrieveMetadata*"
- "*showGraph*"
- "*retrieveSchema*"

The "*listContent*" operation is responsible for obtaining a list of all content currently stored in the repository. It has two optional parameters, "*offset*" and "*limit*" that are used to support pagination of the available content list, while the third one, the common "*contentHandler*" parameter can be used to hint that the list should, if possible, only contain content processed by the specified repository module.

The "*showContent*" operation is primary entry point into the content itself, from the point of view of the repository. It provides an HTML view (be it a complete page or a partial page for embedding uses) that represents the requested content, specified by the "*id*" parameter as seen by the repository. By default the view is generated by the repository module that originally processed the content, or by the core repository module if the repository module is no longer available and it might include some of the metadata associated with the content, depending on the view generator. Additional views may be available from other repository modules, that can be requested by using the optional parameter "*contentHandler*". It should

Operation	Method	Mandatory Parameters	Optional Parameters	Description
listContent	GET		offset, limit, contentHandler	Lists the content available in the repository
showContent	GET	id	contentHandler	Displays the requested content as part of an HTML page.
retrieveContent	GET	id	contentHandler	Retrieves the content in its original format (determined by the underlying content handler)
showMetadata	GET	id	contentHandler	Displays the requested content's metadata only as part of an HTML page
retrieveMetadata	GET	id	outputFormat, contentHandler	Retrieves the content's metadata in the requested format (if possible)
showGraph	GET	id	outputFormat, contentHandler	Displays the content's relations in a graph
retrieveSchema	GET		contentHandler	Retrieves the organisation schema of the personal digital repository or of one of its modules

Table 3.3: Summary of content access operations

be noted that the content itself may still be available from in its original context. If that is the case, the generated view should serve as a gateway to that original context.

The "*retrieveContent*" operation is used to obtain a copy of the content identified by the "*id*" parameter, in its "original" form. Depending on the content, determining what the "original" form is can be problematic. For traditional file based content, "original" form means either the collected file or an handle to where it can be obtained, depending if the content has been completely transferred to the personal digital repository or if it resides elsewhere. On the other hand, non traditional content such as messages or posts from the point of view of the user do not have such a canonical representation. An example of this are text messages in the Android operative system. They exist as entries in a system database (mmssms.db), whose access is mediated through (officially unsupported) content providers, yet for users their appearance and available details are dependent on the application used to display those entries, that might even have its own database with additional information about each entry. Likewise, their underlying representation is bound to be different across mobile operative systems. Thus, while remaining conceptually the same for end users, its representation varies across mobile operative systems and even application within the same operative system (though granted its basic building blocks are all the same). Perhaps the most obvious solution for the representation conundrum is for the personal digital repository to delegate to the module that originally dealt with the content's ingestion what the original form of the content should be. This solution has an obvious risk, as the repository module that dealt with the content may become unavailable, which in turn will require a fallback strategy. Such strategies may come in the form of allowing other repository modules to attempt to provide their version of how that content should be represented, with the use of the optional parameter "*contentHandler*", or having the core repository module providing a last resort representation based on a serialised version of the content generated from its semantic storage representation.

The "*showMetadata*" operation serves as a complement to the the "*showContent*" operation. It is intended to obtain an HTML view (again, either a complete page or a partial template for embedding purposes) that represents the metadata collected about a given piece of content, identified through the "*id*" parameter. Different views, with different levels of detail might be obtained by requesting a specific content handler to generate the view using the

optional "*contentHandler*" parameter. The metadata counterpart to the "*retrieveContent*" operation is the "*retrieveMetadata*" operation. As its name indicates, it is used to obtain a copy of the metadata directly associated with a given content piece, identified by "*id*" parameter, in format specified by the optional "*outputFormat*" parameter. In this operation the optional "*contentHandler*" parameter can be used to determine the module that will actually generate the metadata representation (if the repository module supports the requested content). This allows the personal digital repository to remain flexible, in the sense that different repository modules can provide different levels of detail, omitting or including certain metadata pieces accordingly to its intended use, or support different output formats. Nevertheless the source of metadata is the personal digital repository itself, and thus regardless of the existence of other repository modules, the core repository module itself must be able to generate a serialised version of the metadata it has on any given object, though it should be noted that it will use the underlying scheme already stored, and will not perform any type of scheme translation (which other repository modules might provide). Recommended output formats for the core repository module include JavaScript Object Notation (JSON) and XML.

The "*showGraph*" is an oddball operation. It is intended to provide a visual representation of how interconnected a content piece identified by the parameter "*id*" is. The optional parameter "*outputFormat*" controls the output format of the generated graph, which conventionally should be an image file. Like with the "*retrieveMetadata*" operation, the parameter "*contentHandler*" allows to direct the request towards a particular module, that may support different output formats, for instance outputting not the conventional static image but the data in a format suitable for use with a interactive visualisation library such as Data-Driven Documents [114].

The "*retrieveSchema*" operation is used to obtain a copy of the underlying organisation schema of the personal digital repository's shared context. This can serve as a guide for those who want to want to create new repository modules. This operations has an optional "*contentHandler*" parameter that can be used to specify the module for which the schema should be retrieved, thus exposing the internal schemas used by the requested repository module (and that work as extensions of the core repository organisation schema).

The "*/search*" URL binds together operations that deal with content search. Operations in this group are not allowed to affect the repository's state, though in a slight departure from strict REST design, they may use the POST method if the need arises. This provides an alternative route to perform queries that approaches the limits of what may be practical to do with parameters encoded in the URL itself. The group provides a proxy for the following operations, whose summary can be found in Table 3.4:

- "*search*"
- "*searchSparql*"

While searches are always context dependent, the "*search*" operation in a personal digital repository is even more so. In this operation the core repository module acts as a proxy for parametrised queries that are intended to be passed to search capable modules. While most search capable modules will define their own query language, there are some optional parameters that can be counted upon when performing these kind of queries. These are the "*query*" parameter, that represents the primary query being sent to the repository and whose syntax is dependent of the underlying implementation, the "*offset*" and "*limit*" parameters used to support result pagination, and the titular "*contentHandler*" used to hint to which

Operation	Method	Mandatory Parameters	Optional Parameters	Description
search	GET, POST	query	offset, limit, contentHandler	Performs a text based query
searchSparql	GET, POST	query		Performs SPARQL based query. Requires additional permissions from the repository.

Table 3.4: Summary of search operations

content handler the query should be sent. Once again, if the hinted module is missing or invalid, the core repository module can step up and act as search provider of last resort, allowing queries that target the common fields that compose a base digital object.

The "*searchSparql*" operation differs from the the previously described "*search*" operation. While the regular search can target specific content handlers, and its form and query language is determined by them, the "*searchSparql*" targets the personal digital repository's shared context and its query protocol and syntax is fixed, using the SPARQL specification, thus providing an operation to query the shared context's underlying semantic data. The query itself can be specified using the mandatory "*query*" parameter, and it must be a well formed SPARQL query. The output format can be controlled by an optional "*format*" parameter, with support for various formats dependent on the underlying query engine. Typical formats specifiers include "*XML*" to obtain the query results in the SPARQL Results XML format and "*JSON*" to obtain the results in the SPARQL Results JSON format, with the personal digital repository adding the specifier "*HTML*", to obtain an HTML representation of the results suitable for visualization with a browser. Queries performed by this operation are not intended to have side effects (i.e. to modify data present in the personal digital repository, and as such the exposed endpoint should be of the 1.0 variety that does not have support for content insertion or modification). Despite this, having unfettered access to the underlying sharing context comes with the very real possibility of exposing potentially unwanted content (or worst, inferred relations between content pieces) to the outside world. Short from pre or post processing the queries to ensure that no unwanted results are expressed, the only other potential mitigation measure that can be taken if for external clients using this operation to require an additional permission to actually perform it.

The "*/auth*" URL binds together operations that deal with user and agent authentication and authorisation. As mentioned in the previous subsection, security operations are the sole domain of the core personal digital repository module. The group provides the following operations, whose summary can be found in Table 3.5:

- "*login*"
- "*logout*"
- "*registerClient*"
- "*registerStatus*"
- "*authorise*"
- "*token*"

The "*login*" and "*logout*" operations serves respectively to authenticate a client while establishing a session, and to terminate the previously established session. Authentication schemes, and thus parameters for the "*login*" operation are implementation dependent, though typical parameters for it may include "*username*" and "*password*" combinations. These operations

Operation	Method	Mandatory Parameters	Optional Parameters	Description
login	POST		Implementation dependent	Initiates an interactive session in the personal digital repository (intended for human users)
logout	POST			Terminates the session of the current user in the personal digital repository (intended for human users)
registerClient	POST	toolId, scope, name, description	callbackUrl	Initiates the registration process for content collection tools. This has to be manually confirmed by the repository owner.
registerStatus	GET	registerId		Returns the state of the registration process of a content collection tool
authorise	GET	Implementation dependent		Provides a code that can be exchanged for a token after the content collection tool has identified itself.
token	POST	Implementation dependent		

Table 3.5: Summary of authentication operations

are used solely for interactive sessions established by human users, while other agents such as content collection tools should establish their identity through other means.

As previously mentioned, not all personal digital repositories will rely on the same set of content collection tools, and the content collection tools used throughout time will change, with new ones being added, and other ones being deprecated accordingly to the needs of the repository owner. A content collection tool that is invaluable in some scenarios, might be useless, or a downright threat in others. This means that there is the need to dynamically control which tools can access a given personal digital repository, with the final decision being done by the repository owner. The "*registerClient*" is the operation through which a content collection tool can request access to interact with the personal digital repository. It is invoked using the POST method, and implements a derivative of the previously mentioned OAuth dynamic client registration protocol, with four mandatory parameters. For this operation mandatory parameters are "*toolld*", that represents the identifier by which the content collection tool will register with the personal digital repository; "*scope*", a list that represents the permissions that the content collection tool requires (for instance access the personal digital repository's shared context); "*name*", that represents a friendly name to be presented to the repository owner when requesting its authorisation to access the repository and "*description*" that represents a short text to be presented to the repository owner describing the purpose of the content collection tool. There is an additional optional parameter, "*callbackUrl*" that specifies the desired URL to where clients should be redirected after identifying themselves to the personal digital repository, with other additional parameters being implementation dependent in order to support other OAuth functionalities (such as different authentication schemes). Unlike the OAuth dynamic client registration protocol, this operation does not return an answer immediately, instead returning an identifier that can be used to know the status of the registration request, as to give absolute control to the repository owner, the process can not be fully automated.

The "*registerStatus*" operation complements the previously described "*registerClient*" operation, being intended as the method by which the content collection tool might inquire

the personal digital repository regarding their registration request. To do this tools use the mandatory parameter "*registerId*", to send back the identifier originally obtained from the "*registerClient*" operation. Possible outcomes of this operation include receiving a message with the current status of the registration (i.e. if it is pending, or has been denied by repository owner), or in the first contact after the registration been approved, a message with the corresponding status plus additional data to access the repository, such as the assigned client identifier ("*clientId*") and client secret "*clientSecret*", akin to a passphrase for content collection tools. It should be noted that more information might be returned, depending on the underlying support for the OAuth protocol.

The "*authorise*" operation is intended as one of the initial entry points into an OAuth protected system. Content collection tools send a request to this operation in order to identify themselves and obtain an access token, used in subsequent requests until it expires, or a code that can be exchanged to an access token using the "*token*" operation, depending on the underlying OAuth implementation. The same is true for the required parameters to perform this operation. It should be noted that operation is usually interactive, and requires the user to actively login into the personal digital repository and authorise the content collection tool to act on his behalf. After successful doing this step, the content collection tool can obtain tokens using an automated operation.

The "*token*" operation is the other entry point into an OAuth protected system. This operation also allows the content collection tools to obtain tokens that can be used to access protected resources on behalf of a personal digital repository user (usually its owner). Unlike the "*authorise*" operation, this one is automated and assumes that said user already gave its consent for the tool to act on his behalf. Failure to do will result on failure to obtain a token. This operation can also be used to refresh a previously obtained access token that is about to expire.

Finally the "*/pmh*" URL binds together operations that deal with metadata retrieval through the OAI-PMH protocol operations (such as "*GetRecord*"). The main rational for the inclusion of support for this protocol in a personal scenario is to facilitate eventual repository-to-repository communication using an already established protocol. There are however some caveats to the implementation of the OAI-PMH to adapt them to personal scenarios. The first one is that like any other operations in a personal digital repository, use of the operations defined by the OAI-PMH protocol is restricted to authorised tools only. This is done since content present in a personal digital repository should be considered private by default. A side effect of this is that regular harvesters, that are unaware of this restriction will not be able to use this endpoint. A second caveat is that since it is primarily aimed at repository-to-repository communication, its implementation is optional and might be completely deferred to a complementary repository module (though still mediated through the core repository module). Regarding the operations provided under this URL, one should refer to the OAI-PMH specification [52] for further details regarding their parameters and working mode.

Common Digital Object Composition

As content is gathered and placed into the personal digital repository, it is highly likely that it will come in a myriad of formats that can be described by multiple, often concurrent, metadata standards. Yet, regardless of input format, or metadata standards used, there is a clear need for a common core that binds the input content and associated metadata into a

coherent unit that can be effectively tracked and managed by the personal digital repository. The union of the content itself with its associated metadata and administrative metadata is what forms the digital objects that compose the catalog of the personal digital repository. The elements that make up administrative metadata are fixed, which means that they are the same regardless of content type, while the elements that form the content and its associated metadata are allowed to differ, even between two content pieces of the same type. Fixed administrative elements are not aligned with any particular norm (such as DC), as their primary purpose is to support the operations of the personal digital repository. On the other hand, the elements of the object's metadata can (and would preferably) be aligned with existing norms, though ensuring such alignment is ultimately the responsibility of the module that handled the object's submission in the first place. The fixed elements that comprise the administrative metadata of a digital object are:

id The unique identifier assigned by the personal digital repository to a digital object. The actual implementation of the personal digital repository defines the scheme used to generate these identifiers as long as it remains consistent regardless of the object type, though they should strive to use primarily characters that are URL-safe.

ingestionDate The date, accurate at least to the second in which the content was initially placed in the personal digital repository.

modificationDate The date, accurate at least to the second in which the content's metadata was last modified.

initialHandler The repository module through which the content passed when it was initially ingested.

contentFormat An identifier that represents the underlying format of the digital object, preferably one using a standard scheme such as PRONOM [115] or Media Types [116], which are also known as Multipurpose Internet Mail Extensions (MIME).

contentType An identifier that represents the type of content associated with the digital object for the use of repository modules. Repository modules can register themselves as handlers for different content types, and can register content types with the repository. This differs from "*contentFormat*" identifier since it is an internal identifier that is independent of the underlying content format, and instead provides a basic identifier for the concept that it is meant to represent (for instance a social media post instead of plain text) as was seen by either the repository module or the content collection tool.

revision An identifier that represents the revision of the digital object. Personal content can be collected in various stages of its evolution, with each subsequent stage representing both an evolution of the previously collected revision as well as independent content piece by itself.

previousRevision An optional field that contains the identifier of the digital object that represents previous revision of the content piece.

nextRevision An optional field that contains the identifier of the digital object that represents the next revision of the content piece.

checksum A verification code calculated by cryptographic hash function chosen by the underlying implementation of the personal digital repository. It can be used to verify the integrity of the digital object's content part. This element may be supplied by the content collection tool or be left blank, with these options being provided to deal the scenarios where the content collection tool only gathered metadata and not the content itself.

checksumAlgorithm A string that identifies the algorithm used to calculate the content's "*checksum*" element. Like the "*checksum*" element, this one can also be supplied by the content collection tool or be left blank.

originalContext An optional field that contains an handle capable of leading the user back to where the content was originally located. Used primarily for when the content is neither application nor device bound.

contentStorageType Indicates how the content is stored within the personal digital repository.

name An optional field that contains an human readable "friendly" name derived from the content (for example a file name, or a title), mainly for fallback text based queries performed by the core repository module.

description An optional field that contains a textual description of the content for fallback text based queries performed by the core repository module.

metadata An optional field that contains object bound metadata. It serves as an aggregation structure under which additional metadata properties can be placed. These properties should, if possible, be represented using their original schema or elements (for instance DC, MPEG-7, EXIF) as sent by content collection agents or extracted by the core digital repository module when acting as a fallback for generic content.

Regarding content representation and storage, there are three possible scenarios that can happen. The first scenario is that the content can be stored directly with the metadata components; the second scenario is that the content is already present in the personal digital repository and the third scenario is that content has yet to be placed in the personal digital repository itself. The first two scenarios will lead to the inclusion of an extra "*content*" element, whose interpretation differs according to the scenario. In the first scenario the content is stored inline with the metadata and administrative metadata, which means that the "*contentStorageType*" element of the administrative metadata will have a value of "*inline*" and there will be an additional "*content*" element present to actually hold said content. The internal representation of the "*content*" element is dependent on the repository module that originally processed the content at ingestion time. This representation mode should be used for content which is primarily textual in its nature and without an well established representation (such as text messages). In the second scenario content is stored by one of the available content storage modules. As such the "*contentStorageType*" element of the administrative metadata will have a value of "*internal*". In this mode the additional "*content*" element is a structure that contains a list of module/identifiers pairs that can be used to retrieve the content from the content storage module. Allowing multiple module/identifier pairs, when coupled with multiple content storage modules gives the personal digital repository basic replication capabilities. In the

third scenario content is treated as an external resource. The "*contentStorageType*" element of the administrative metadata will have a value of "*external*" and in this mode there should be an "*contentExternalLocation*" element that contains an handle that will lead the repository owner to the content itself. This element may also be present in cases where the content has been placed in the personal digital repository but is also available from the external source, being that in this case it acts as a link to the content in its original context.

Shared Context

The previously described common digital object is intended to act as a package that can be manipulated as a whole by the repository itself. The only certainty is that it contains administrative metadata that is mainly useful to the personal digital repository itself, with the presence of other, object bound, metadata types being dependent on the repository module that processed the content at ingestion time. This allows the personal digital repository to group into a single administrative collection objects that can be very disparate in their composition, yet this comes at the cost of limiting the default available metadata within that collection. This means that despite being used to store metadata it does not make any provisions to use it to establish relations with other objects present in the personal digital repository other than establishing fields to link to previous and subsequent revisions of a given digital object.

Given that the primary role of object specific metadata is to provide additional information that can guide the personal digital repository owner back to its content, and that metadata itself can be thought as a series of relations between concepts, that can be shared across multiple digital objects, it becomes clear that a personal digital repository requires an additional metadata storage that is not constrained by the bounds of the common digital object in order to take full advantage of the gathered content. To fulfil this role, the personal digital repository establishes a shared context. The goal of the repository shared context is twofold: to used gathered metadata to establish relations between disparate pieces of content and to provide the schema under which collected content is classified and organised. Relations can come in many forms, from explicitly shared metadata items (for instance having a common author or message recipient) to temporal relations (for instance the creation of one object pre-dates the creation of another) to cause-effect ones (for instance the content of a message influenced the modifications done to another content piece) to organisational ones (for instance two content pieces become related by being part of the same collection). Furthermore, in something like a personal digital repository that is expected to gather content throughout the life of its owner, relations between content pieces are expected to change, either due to direct actions of the repository owner (for instance removing a piece of content from a collection) or to circumstances (for instance a drop in exchanged messages due to gradual drift between the repository owner and a friend). Thus in addition to know that the relations existed, it also becomes relevant to track its duration and type.

Since there is the need to store a lattice of content pieces along with the relations defined by their metadata and also to keep track of the type of relation, there is a strong incentive for the repository shared context to be based upon a type of semantic storage, where the stored data is governed by an ontology that ties together the organisation and concepts used throughout the personal digital repository which is described in detail in chapter 4. Additionally, as the construction of the shared context hinges primarily on the collection of metadata, this also provides a strong incentive for the personal digital repository to adopt collection polices

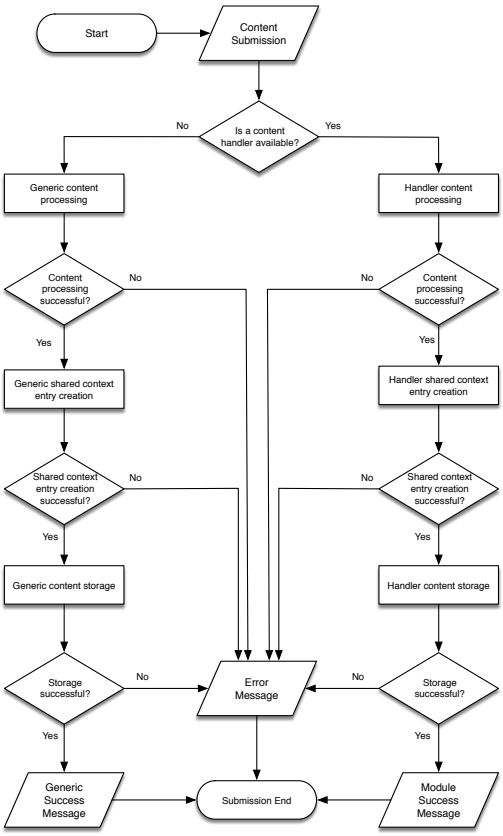


Figure 3.4: Content Submission Process

that prioritise metadata over the content itself, thus further contributing to turn the personal digital repository into a meta-repository.

Content Submission Flow

The content submission process (shown in Figure 3.4) is triggered by issuing a request to the "*storeContent*" operation of the "*/dataManagement*" URL group. The first step taken by the core module proxy for the operation is to determine if it can be directly routed to a specific repository module. It does that by checking for the presence of the "contentHandler" parameter. This parameter is only valid if it matches a registered module's identifier. If the "contentHandler" parameter is missing or invalid the core module proxy attempts to determine the submitted content type from present attachments, matching its type to those that claim to be processed by available modules, otherwise it will be forced to act as a fallback content handler. If an appropriate repository module has been found, the request should be forwarded to it.

Depending on the verb used to make the request, it can be considered a new content submission (POST) or an update to existing content (PUT). Updates should only be made in particular cases, such as when submitting the actual content when the repository had previously only held an handle to the external location of the content, to add additional locations where the content can be found, or to correct erroneous metadata. Modified versions

of already existing content should be treated as new submissions albeit explicitly linked to the previous version at least by their administrative metadata. It should be noted that in this scheme, the power to determine if a submission should be treated as new content or as an update is ultimately held by content collection agents, and as such some care is required in their creation.

When a specific module is chosen to process submitted content it is responsible for creating and filling the digital object's common properties for each submitted content instance (as interpreted by the module itself). It is also responsible for the extraction of additional metadata, part of it might actually have already been supplied as additional parameters with the request by the content collection agent, with this additional metadata being stored as additional fields in the created digital object. If the proxy module is acting as a fallback content handler, the only warranty given is that the common administrative metadata will be filled. Additional attempts to retrieve more metadata are optional and left to the actual implementations. After storing the administrative portion of the digital object, the module is also responsible for placing the content in the personal digital repository shared context, along with the establishment of any explicit relation with existing objects. At the minimum it must create the events that describe the ingestion of the content, thus ensuring that temporal relations between the ingestion of the content can be established within the shared context. This much is guaranteed to happen if the proxy operation is acting as a fallback content handler. Additional relations and properties can be derived from the metadata of the digital object, though some of it might require specialised extensions to be described. Modules are free to provider their own extension to the ontology used to organise the personal digital repository, as long as it remains consistent with it, and that their extension can be retrieved through the "*retrieveSchema*" operation. This last requirement is done to ensure that it can be used in the development of other repository modules. Finally, if the "*content*" parameter represents a multi-part attachment it is the processing module's responsibility to decide if it should be stored (it is conceivable that the module receives a compressed archive that contains the actual objects being submitted, in which case the container might by nothing more than an artefact created by the content collection agent), and if so to use one of the storage provider modules available in the repository. When the "*content*" parameter is not a multi-part attachment, it is assumed that it represents some kind of textual representation that can be, if necessary, stored in the optional "*description*" field of the administrative metadata. Only after this storage step the content submission process can be considered as completed. It should be noted that content in submissions that are handled by the proxy operation is essentially treated as black box. This allows the repository owner to directly submit content even if no module is available to process it, which at the very least ensures that there is a record of the existence of said content (and a copy of it stored in the primary storage provider if it is file based).

3.2.2 Digital Inheritance

Digital inheritance is a complex issue that can not be solved by technological means alone, as its many peculiarities will require the development of a legal framework that is outside the scope of this document. Nevertheless, personal digital repositories can contribute to the subject by means of the policies that they adopt.

As previously stated, unlike physical content, digital content can be perfectly replicated. Theoretically this means that a digital estate can be passed to all of its rightful heirs and legatees as a whole, instead of having to be divided between claimants. Unfortunately, this

only holds true in theory, since in practice parts of the digital estate may be encumbered. While some online services include in their service agreement provisions to deal with the death of the account holder, ranging from turning the account into a memorial (Facebook) [117], provide copies of the account contents (Google's gmail) [92] or simply to delete the content (Apple's iCloud) [118], others don't include any provision at all and actively forbid account ownership changes, which can make it difficult for the rightful heirs to actually retrieve the content. Another related issue that may arise is that not all of the digital content that forms a digital estate might be fully owned by the individual that collected it. Subscription based content, for which individuals acquire the rights to access, but do not own the content is an example of this, as are content pieces that were not legally acquired. To further complicate the matter at hand, the repository owner might simply want to bar some of its heirs from inheriting specific pieces of its accrued digital estate.

Existing approaches to digital inheritance, such as the ones offered by the Cirrus Legacy [119] service deal with content and online accounts on an individual basis. Their basic approach is to act as a digital locker where the credentials for online accounts are stored, ready to be bequeathed to their user's heirs. While convenient this approach is not without flaws. The unfettered access to the online accounts provided by receiving their respective credentials means that it is an all-or-nothing strategy: the designated heir will have complete control over the account for which he has received the credentials, which in itself might be undesirable or outright illegal, it also has means that the designated heir will most likely also have access to content associated with the account that may have never been intended for him, with the only possible mitigation measure being removing said content beforehand (which may not always be possible). Furthermore effective control of an account can only be given to a single heir, while the content itself might be intended to multiple heirs, which is an odd predicament for the majority of digital content, whose chief advantage is that it can be perfectly replicated.

While there is little that a personal digital repository can do regarding digital inheritance of subscription based content other than to know that the repository owner once had access to it (thus being effectively limited to bequeath proof of memories), it can contribute to solve some of the other challenges posed by digital inheritance. Inheritance policies can be applied on a per-content basis, thus reducing (but not removing) the need to explicitly leave behind account credentials. Collected content can be replicated from the personal digital repository instead of directly from online accounts, removing the need to do it manually if there are multiple heirs while at the same time opening the possibility of keeping a record of where the inherited content came from, including it had been previously inherited, if the replication leads it to another personal digital repository. Over time and multiple inheritance cycles, this can be used to establish the content's provenance. The possibility of establishing a multi-level and accurate provenance is in itself a departure from the available digital inheritance strategies that simply provide access to an account, since those can only establish implicit single level provenances (i.e. second level heirlooms can contain credentials for accounts that were previously inherited but they are treated in the same way as first level heirlooms and must be explicitly accessed in order to determine their original owner).

As was previously mentioned, in a personal digital repository establishing who should inherit what is done on a per-content basis. The default policy for every object should be to follow traditional inheritance rules, with the owner's next of kin being granted copies of all content. In addition to this default traditional policy, a personal digital repository must also support four additional inheritance policies: *Not Inheritable*, *Allowed Inheritance List*, *Denied Inheritance List* and *Memory Institution List*. The *Not Inheritable* policy marks a

given content piece as not inheritable, thus denying the default personal digital repository policy while at the same time rendering the content inaccessible (baring physical access to the device where it is stored). The *Allowed Inheritance List* policy serves as white list approach, overriding the default policy and replacing it with a list of potential inheritors. Only those present in the list will be allowed to access the content, being that the list can contain legatees (such as friends or coworkers), though it must also include the direct descents if they are to inherit the content to which this policy is applied. The *Denied Inheritance List* works as a blacklist, essentially preventing those on the list from inhering the content. Though it can be applied to any heir, it is primarily aimed at those that would have the right to inherit by default. Finally the *Memory Institution List* is an opt-in policy that allows the repository owner to leave content to memory institutions. It is used in conjunction with the other digital inheritance policies, and does not override the default policy.

In order to support inheritance polices, heirs and legatees need to have a representation in the personal digital repository's shared context. Default heirs are determined by finding in the personal digital repository persons who have a direct family relationship (for instance sons, stepsons or current spouses) with the repository owner. While the repository owner must be able to specify them (and change their details) if there are none present in the personal digital repository (such as in an initial phase where the repository is empty), it might also be possible to automatically discover them, if for instance they are signalled as such by a content collection tool that operates over social graphs gleamed from social network services, though such a scenario is obviously dependent of the type of content gathered by the repository owner. The bare minimum amount of information (in addition to the relationship) needed to create an heir is an appellation by which he will be known and some form of contact usable by the personal digital repository to inform him of the digital estate's availability. Legatees are created or discovered in a similar way and require similar information (with the addition of a memory institution flag for special purposes), but lack the direct family relationship, though they should be marked with another kind of relationship. As such they aren't automatically entitled to anything unless the repository owner explicitly bequeaths them something using the *Allowed Inheritance List* policy. In addition to having a representation, those involved in the digital inheritance process need to have an account created in the personal digital repository. These accounts serve as one of the means by which the identity of those who have a stake in the digital inheritance process, and can be created (and removed) by the repository owner, and have their own set of access credentials. A personal digital repository can provide to its owner some different options to manage the release of these access credentials, from automated methods (i.e. for instance by sending an email to the heir with the access credentials, either immediately or when the digital inheritance policies are triggered), to manual releases that require the repository owner (or a digital executor) to take actions to inform the rightful heirs of the credentials.

Determining when to trigger digital inheritance policies can be problematic. If timed incorrectly, the release of content will undoubtedly have unexpected consequences. Triggering the release of content too soon is can be considered anything from a nuisance to a full blown security breach and may, depending on the released content, end up having a very negative impact in the repository owner's life or reputation, while triggering it too late may make the content released nearly irrelevant for the heirs (though there is a special case in this scenario, that is when one of the heirs is a memory institution). While arguably there will never be a perfect release strategy, there are three main approaches that can be taken to determine when to trigger a personal digital repository's digital inheritance policies:

Inactivity Policy Digital inheritance policies are triggered if the repository owner is inactive for a set amount of time (defined by the repository owner himself). An inactive repository owner is defined as one that did not accessed the personal digital repository and whose content collection tools did not gathered any content during the defined time span. In order to help preventing (though not eliminate) accidental inactivity, the personal digital repository must attempt to contact the repository owner (for instance using warning emails or text messages) multiple times at fixed intervals when inactivity is beginning to be detected. The risk of resetting the inactivity period by having automated content collection tools feeding back to the personal digital repository the contact attempt messages can be limited by imposing mandatory access token renewal policies that force the repository owner to reauthenticate himself with the repository before issuing a new access token.

Inactivity Policy + Embargo Follows the same rules as the previous approach, but after informing the designated heirs places an embargo in the content for a set amount of time. Digital inheritance policy can still be cancelled or reversed during the embargo if the repository owner becomes active again. This approach is designed with memory institutions in mind, so that they can be notified when they would be receiving some content and can take appropriate measures (such as securing the resources to keep the personal digital repository active) while at the same time ensuring that content is only released after a waiting period (for instance to preserve the repository owner's reputation).

Digital Executor Instead of relying on automated procedures, this approach introduces a third party whose sole responsibility is to trigger the digital inheritance policies. A digital executor is not authorised to change any of the defined inheritance policies and is forbidden to access the content contained in a personal digital repository. Instead he can at any time access the personal digital repository and trigger the heirs notification process. To prevent the system from being abused there is an short embargo, over which the personal digital repository attempts to contact its owner so that he can cancel the process. It should be noted that the digital executor himself must have a previously created account in the personal digital repository as well as the associated credentials.

Following triggering of the digital inheritance policies, the next step is the delivery of inherited content to heirs, with two possible scenarios depending if the heir has an accessible personal digital repository of its own or not. If the heir has a personal digital repository of its own that can be accessed, then the core repository module is responsible for packaging each of the inherited content pieces, associated metadata and inheritance metadata and submit them to the heir's personal digital repository. Since the origin repository will effectively act as a content collection tool, it will need to go through the previously described tool registration process before it is able to interact with the heir's personal digital repository. If the heir does not have a personal digital repository of its own then the origin personal digital repository will need to act as one of the more traditional digital inheritance services, allowing the heir to directly retrieve content without any of its associated metadata or inheritance metadata. In this scenario the only advantage of using a personal digital repository for digital inheritance management is the ability to define inheritance rules on a per-content basis, since the inheritance metadata that could contribute to establish the provenance of a given content piece would be left behind. A personal digital repository can also act as credential safe for online accounts. If it possesses a representation of an online account that includes access

credentials, that representation can be treated as regular content and subjected to inheritance policies. This will obviously remove the benefits of having per-content approach to digital inheritance for content that came from that account, but will provide a useful fallback for when heirs do not have have an accessible personal digital repository. Furthermore, even with the inclusion of a personal digital repository, in both scenarios it should be noted that the time that heirs have to retrieve inherited content might be limited by external factors. Payment owned for domain registrations or to third party service providers might limit the content's availability, particularly if the repository owner choose to self-host his repository and did not made any contingency plan to ensure the repository's survivability for long enough after his demise.

3.2.3 Repository Deployment

There are two primary scenarios for the deployment of the personal digital repository, each of them with their own set of advantages and disadvantages, with other deployment scenarios to be considered hybrids between the primary approaches.

The first scenario is to follow the Do It Yourself (DIY) route, in which the personal digital repository is deployed by the users themselves. This scenario has more in common with the deployment of a traditional digital repository, with the notable difference that instead of an organisation it will be a single individual that will have to deal with all the administrative and technical burdens that might arise from said deployment. A notable advantage of having individuals deploying their own personal digital repositories is that content, collection agents and the personal digital repository will be under the control of the repository owner. This can curtail the number of third parties that will have access to the potentially sensitive information contained in the personal digital repository, especially if the repository owner takes upon itself the burden of maintaining its own infrastructure (i.e. primarily the hardware that supports the personal digital repository, but also the personal digital repository software). Each piece of the infrastructure outsourced to third parties transforms this scenario into an hybrid and potentially introduces more points through which information can be leaked without the knowledge of the repository owner. The most obvious disadvantage of this scenario is that the repository owner must ensure all of its maintenance, both from an administrative and a technical point of view. In a digital repository designed to act as a web application, one must take into account not only immediate maintenance priorities, such as hardware upgrades but also long term connectivity provisions or domain registration and maintenance. When coupled with the possibility of wanting that some of the content present in the personal digital repository to be inheritable, these provisions will have to be taken in advance. If not carefully planned, they may hinder any automated inheritance system that the personal digital repository might have in place, potentially depriving heirs of content that would be rightfully theirs. On the other hand, just as interests change over time, the willingness to fully maintain a personal digital repository can change over time, especially about the time it will require more maintenance effort (such as hardware upgrades). This might lead the owner to abandon its repository, effectively undoing any good that it might have previously accomplished in terms of preserving his digital history. On the other hand, by maintaining complete control over the entire stack, even "abandoned" repositories have the chance to be rediscovered and partially fulfil their role in digital inheritance schemes, as long as the physical hardware is still around and in working condition.

The second scenario is to follow the route of Software as a Service (SaaS) and outsource

the bulk of the infrastructure needed to maintain a personal digital repository to a third party provider. The obvious trade-off is that repository owners become repository users, relinquishing absolute control over the entire stack in favour of convenience while still deciding what content is placed into the personal digital repository by retaining full control over the collection tools used to gather it and by choosing the modules that will support the core repository. In a SaaS deployment scenario the responsibility for long term "headaches", be them administrative (such as domain registration) or technical (such as hardware upgrades) in nature can be shifted to the service provider, potentially alleviating the burden placed upon the repository owner and thus contributing for its decision to keep using the system. Furthermore, service agreements can include the necessary provision to maintain a personal digital repository accessible for a sufficient period of time that allows automated inheritance strategies to take control and deliver the repository's contents to the rightful heirs or to a memory institution. The introduction of a third party that maintains the personal digital repository infrastructure can bring with it trust issues. While some of those may be solved by including clauses in the service agreement that forbids the service provider from directly accessing or using the content present in a personal digital repository (barring any unavoidable provisions made for access by law enforcement agencies), some such as the long term stability and viability of the service provider cannot be effectively controlled. Any service provider willing to build an infrastructure for personal digital repositories must take into account that individual repositories must be isolated from each other. This is both to protect the privacy of the repository owner, and to ensure that there is no unintentional cross referencing between different shared contexts belonging to different personal digital repositories or between their digital objects. This goal can be achieved for instance, by packaging the personal digital repository as a software appliance that can be run in standard physical or virtual containers, providing each user with their own database and triple store instance, therefore respecting the personal part of the digital repository.

Both deployment scenarios have their advantages and disadvantages, being that the first one is more likely to yield a truly independent personal digital repository yet is much harder to maintain, while the second scenario is arguably much more convenient for end users though it relies on the existence of a third party service provider. Since one of the criticism pointed at traditional digital repositories in the previous chapter is that they often require specialised personnel to support them, and that in personal scenarios it is arguably more likely that users will only have minimal computer use knowledge, a system of personal digital repositories will have better chances to succeed if it is offered as a service, instead of being a DIY kit. Furthermore, by being offered as a service, individual personal digital repositories can, if needed, move towards formal federation.

3.2.4 Actors

The previous subsection indicate out that in spite being in a personal scenario, the repository owner is by no means the only actor that interacts with the personal digital repository. Potential interactions can come directly from the repository owner himself, from content collection agents, external visitors, heirs, legatees and digital executors.

The main interactions should come from the repository owner and the automated content collection agents. Secondary interactions should come from external visitors (if the repository contains any content deemed public by the repository owner) an by those involved in the digital inheritance processes. Traditional repository roles of producer (i.e. all those who

are responsible to provide content for the personal digital repository), consumer (i.e. all those who access the system in order to retrieve content from it) and manager (i.e. all those responsible to set the policies that determine the behaviour of the personal digital repository) are still applicable to those who interact with the personal digital repository, though with some nuances. The repository owner can fulfil all three roles, as he can procure and supply content directly to the personal digital repository, consult the gathered content and he is the only responsible, either directly or indirectly for setting the repository policies. He is also the driving force behind the content collection agents, that also supply content to the repository (and thus serve as producers) and access the stored content (thus serving as consumers). Since the dual purpose content collection tools are not acting out of their own volition but on behalf of the repository owner, one could be tempted to make the case that they by themselves are not producers and consumers, but fulfil an additional role of mediators, standing between the actual content producers or consumers and the repository. Yet, the dual nature of content collection tools, that provides a convenience service in exchange for the content to place in the personal digital repository also means that in order to provide said service they need to access, and if needed transform for presentation purposes the content present in the personal digital repository, thus earning them the role of consumers in their own right. Furthermore, they collect and submit content to the personal digital repository, content that they need in order to provide their "cover" service, content that could otherwise go uncollected and that can be considered to be collected only with implicit (rather than explicit) authorisation from the repository owner, thus earning them the role of producers also in their own right.

Secondary interactions should come from external visitors and those involved in the digital inheritance process (heirs, legatees and digital executors). Even though they are separated by content access policies, external visitors, heirs and legatees can share the role of content consumers. After all they do not contribute directly with content to the personal digital repository (which would make them take the role of producers) and should not be able to directly influence the repository's policies (which would make them take the role of managers). Digital executors' interaction with the repository should be extremely limited. They do not need to access the content in order to trigger the established inheritance policies and thus they do not take the role of consumers. They also don't contribute with content for the repository, thus preventing them from having the role of producers. Finally they do not have the power to directly change or determine the repository policies which also excludes them from the manager role. As the task they need to accomplish is more akin to the daily administrative tasks needed to run the repository, that are usually condensed as responsibilities within the repository instead of a interaction roles, digital executors are also not given an interaction role.

3.3 Comparison with OAIS

There are some issues that prevent a personal digital repository from aligning perfectly with the goals and requirements defined in the OAIS reference model. The definition of an OAIS states that "*[a/n OAIS is an Archive, consisting of an organization, which may be part of a larger organization, of people and systems that has accepted the responsibility to preserve information and make it available for a Designated Community*" [13]. This statement of purpose is not completely applicable to a personal digital repository, since while it is undoubtedly an organisation of people and systems (even if the only person directly involved is the repository

owner) and it is possible to establish a designated community (composed by the repository owner himself and the potential heirs and legatees), the repository owner may not have as its goal the preservation of information but merely to keep track of it, with preservation being a side effect from this goal. The word "*Archive*" implies that there are policies in place that restrict the availability of the content, something that a personal digital repository also adopts, yet it usually also implies that there is a coherent, curated organisation scheme in place, which is something that does not necessarily happen in a repository (that may be something as simple as an disorganised pile of papers in box). While a personal digital repository obviously will not be void of organisation, the one in place may not achieve the quality standards required to qualify as fully fledged archive.

As mentioned in the previous chapter, to be compliant with the OAIS model an archive must accept and fulfil a set of six responsibilities. While a personal digital repository does not have any predictable issue with the first (i.e. to negotiate and accept information from producers), or the third (i.e. determine the designated community) responsibilities the remaining four are only partially fulfilled. Regarding the responsibility to obtain sufficient control over the information, the content contained in a personal digital repository, though expected to be primarily created by the repository owner himself may also include content acquire from third parties. While said content has a reason to be included alongside the other assets in a personal digital repository, since it was at some point important enough for the repository owner to acquire it and might have had an influence on him, it might not be possible to ensure that the repository owner has necessary permissions (for instance at a copyright level) to enable the creation of a backup consigned to the personal digital repository. Furthermore, there may be instances of content that for which the repository owner only owned a license which will only appear as external references in the personal digital repository and as such are also encumbered. Another issue arises with the responsibility of ensuring that the information is understandable by the designate community without expert assistance. The designated community for a personal digital repository presumably is composed of the repository owner, which can be assumed to understand the content gathered in his repository and possibly the group of designated heirs and legatees. It is unlikely that an average repository owner will feel the need to explain or even to discuss the inheritable content with each of his heirs, particularly if the bequeathed content includes professional documents that elaborated by the repository owner. These would have value for the direct heirs primarily as mementos instead of because of their intrinsic content. Additionally, while the personal digital repository is has a shared context that can be used to provide the background information about collected content, is is by no way complete. Events that leave no digital trace can still influence the repository owner, leading to the appearance of clusters of content that might appear unrelated with the rest. Without access to this external information, that sometimes might even be only of the knowledge of the repository owner himself, there will always be gaps in the information needed by those in the designated community to fully grasp the context of a given piece of content found in a personal digital repository. Regarding the responsibility of ensuring that there should be no ad-hoc deletions, in personal scenarios one can argue that all deletions are ad-hoc. A personal digital repository is supposed to represent its owner's vision take on the world. Events that happen in the daily life of the repository owner might lead to content removal requests by the repository owner, in order to excise part of his own past. While the personal digital repository can include measures to delay, or mitigate the effects of spurious content removal requested by the repository owner in the heat of the moment, in the end, the need to reflect the repository owner's point of view (and not necessarily of retaining all

facts backed by content) in personal scenarios becomes more important than the preservation of unwanted content. Finally, the responsibility of ensuring that disseminated information is marked as being either a copy or traceable to the original to provide a chain of provenance is once again unfeasible in a personal scenario. Repository owners have the legitimate expectation that content placed in the personal digital repository under the guise of being used by other services is kept in the same conditions as when it arrived. Thus visibly modifying it in any way (for instance by adding watermarks) when it is accessed through the personal digital repository, particularly if it is expected to be later reused or restored as part of the services offered can have the negative effect of making the repository owner feel betrayed and that he is not in complete control of the system. In personal scenarios breaching the repository owner's trust is akin to an invitation to stop using the repository and its associated services and content collection agents.

Functional entities (i.e. those that provide services) in the OAIS model that deal with content can be mapped onto the responsibilities that were identified as being of the domain of the core module of a personal digital repository, as seen in Table 3.6. The mapping isn't complete since not all of the services defined by functional entities have to be digital in nature or are applicable to personal scenarios. The "*Administration*" functional entity lack of mapping is an example of both of these issues, since in OAIS it is responsible for activities that are not digital in nature (for instance the maintenance of hardware) or that are not translatable to personal scenarios (customer support or negotiating the rights to content). Not having a direct mapping does not mean that applicable tasks do not have to be done, only that the personal digital repository can not provide any support with those matters. The "*Preservation Planning*" functional entity is also not directly mapped. Long term preservation requires a detailed analysis of the content at hand in order to avoid digital obsolescence. Such analysis must be a deliberate act, that eventually leads to the establishment of preservation plan and road map and requires a very specific set of skills, that the average repository owner is not bound to possess. While the personal digital repository might offer tools via modules that for instance, support format migration, without a coherent plan these can end up doing more harm than good. Thus the primary contribution that a personal digital repository can do to further the goal of long term preservation is actually to be able to gather non traditional content that would otherwise be lost. Further preservation activities should be deferred to memory institutions if they are part of the legatee list. It should also be noted that the *Data Management* functional entity, that deals with the information system that supports the archive is mapped to the content storage responsibility since metadata storage (and with it control of the information system) has been folded into that responsibility. The different information packages suggested by OAIS to be used for specific functions can be mapped to the personal digital repository nearly directly, with the SIP being roughly equivalent to the collected information provided by a content collection agent, the AIP with the digital digital object stored in the personal digital repository and DIP to the output of the content modules (or the core module) when an digital object is retrieved from the personal digital repository.

The slight mismatches between OAIS functional entities and the personal digital repository mainly arises since OAIS attempts to codify the activities required to run an archival system, while the personal digital repository is more concerned with the means that can be used to achieve them. These activities include both policies that need to be adopted and interactions between external actors that are either outside the scope of a personal digital repository, or too cumbersome to ensure in personal scenarios. In fact the need to adopt more lenient policies for personal scenarios (e.g. relaxed content ownership claims/rights) the root cause

OAIS Functional Entity	Personal Digital Repository Responsibility
Ingest	Content Management
Archival Storage	Content Storage
Data Management	Content Storage (Metadata Storage)
Administration	-
Preservation Planning	-
Access	Content Retrieval
Common Services	Security

Table 3.6: OAIS to personal digital repository mapping

that leads the personal digital repository, without any outside intervention, to only be able to partially fulfil four of the six responsibilities required for OAIS compliance. Thus, while a personal digital repository should not be considered OAIS compliant in the typical use case, should the repository owner choose of his own accord to adopt and perform the necessary policies and negotiations with external actors, and to use modules that provide functionality to further support management activities (for instance report generation) this statement has to be re-evaluated.

3.4 Chapter Conclusions

This chapter describes a possible architecture for a system of personal digital repositories. The proposed architecture attempts to solve the issue of collecting personal content by using content collection tools that, in addition to content gathering, also provide one or more services that a repository user might deem useful, with the personal digital repository ostensibly placed as a back end storage and support for those services. The end objective being not to attempt that gathered content remains usable (i.e. for instance in a format that is understandable by future technology) but to increase its chances of surviving and eventually being rediscovered.

While bidirectional content collection tools (i.e. those that both collect and retrieve content to provide their additional services) can be used in rediscovery events, it is far more likely that the personal digital repository itself is going to be used for this purpose. Given its role as an hub for the collected content, a personal digital repository can offer its owner a view not only of the content itself but also of how it related with other available content. In essence, with enough content collection tools collecting multiple content types, particularly non traditional content, a personal digital repository has the potential to capture the shared context that surrounds individual content pieces, thus fostering an environment more favourable for rediscovery events that the one possible in highly specialised, though bidirectional, content collection tools. The shared context allowed by the personal digital repository by no means should be seen as the definitive context for a given content piece. Collection failures due to unforeseen events (for instance network errors), relevant information that is only known to the repository owner and is not in a digital form (and thus can not be gathered and placed onto the repository), or simply the evolution of the context in which the content was inserted after it was collected (for instance new comments for social media posts) can all make the shared context incomplete. This perceived weakness can be countered by having the personal

digital repository act as a gateway to the original content in its original context whenever possible. This solution has the additional effect of also helping to promote rediscovery events, this time in the content's original context (if still available). The dichotomy between acting as a gateway and actually holding the content itself is addressed by the content collection policies, which prioritise gathering metadata (and as such context building) over actually content gathering, which should be done primarily if said content is deemed as risk, or is needed to provide the content collection tools enhanced services. In essence, metadata oriented content gathering polices brings personal digital repositories closer to being a meta-repository. Nevertheless a meta-repository is still a repository and as such it requires a defined set of services and interfaces, that were described throughout the chapter.

A personal digital repository's role as an hub can also be exploited to support digital inheritance. Digital estates can be highly dispersed, with parts attached to service accounts whose terms of service actively hinders the rights of heirs and legatees. Bequeathing the credentials to those accounts is a not an optimal solution, as there is only an account and there might be more than one heir or legatee, thus (and without any other measures) effectively denying the chief advantage that digital content has over physical content (i.e. that it can be perfectly replicated) when it comes to digital inheritance. Furthermore the repository owner might not intent that all of the content associated with those accounts to be given to a single person. The introduction of a personal digital repository allows that content that has effectively been collected to be handed the designated heirs and legatees, while at the time giving the repository owner the possibility of establishing more granular controls (i.e. per content instead of whole accounts) of who will inherit what. If the heirs have their own personal digital repository, the digital inheritance protocol will not only transfer the raw content but also the information about where it came from and who it previously belonged to, thus contributing to establish that content's provenance. In order to participate in the digital inheritance process, a personal digital repository must remain active and accessible after its owners' demise. Given the premise that a personal digital repository can be fielded by an individual, this may not be an assured scenario. While some individuals and early adopters would undoubtedly be attracted by the possibility of having complete control over all aspects of their personal digital repository (from hardware to software to negotiating support services contracts such as Internet access or domain name registrations) others, particularly those less technically inclined probably would not be thrilled for having to deal with such minutia. To prevent a high churn rate and its associated information loss among these users, personal digital repositories could be offered as a service, essentially trading absolute control for convenience and opening the possibility of having federated repositories.

Finally, it should be noted that while a personal digital repository system may not be, "out-of-the-box" OAIS compliant, most of the issues and incompatibilities are political in nature. Should the repository owner choose to address them and assume those responsibilities, there is nothing that prevents a personal digital repository from becoming OAIS compliant.

Chapter 4

Repository Shared Context

As seen in the previous' chapter content cycle, production and acquisition of personal content can be influenced by many factors. From daily events that spur the creation of new content to collaboration or recommendations provided by co-workers and friends; from the feedback effect provided by revisiting and re-evaluating previously acquired content to simple temporal coincidence it is likely that content that is either accrued or personally produced will be, to a degree, interconnected. Personal digital repositories can explore the content that they held in order to offer their users a shared context: a canvas where it can place the connections between apparently disparate content pieces, individuals and chronology can be made apparent. Different types of content, or even the same type gathered by different content collection tools will wield different pieces of metadata that can be used to construct a shared context, and which in turn also contributes to the personal digital repository's organisation. The diversity of metadata that can be gathered comes with the caveat that some of it might be the same, but with a different representation. Additionally since the shared context is a canvas, it is always possible that the repository owner wants or needs to add information to it, be it connections that can not be automatically made by the personal digital repository or because it wants to complement organisation scheme with its own subject based approach that better reflects its interests. Without any addition intervention, in a worst case scenario the different metadata representations and the user defined organisation scheme might end up being either ambiguous or strait up incompatible with each other, thus hindering the development of the repository's shared context.

While it would be tempting to address the problem by having the repository owner to (roughly) design its preferred organisation scheme as a first step of the creation of a personal digital repository, such an approach would likely produce organisation scheme that would be incompatible with each other, thus effectively hindering interoperability between the personal digital repositories of two individuals. Furthermore it can be difficult for individuals to predict how their interest would evolve, being that this evolution will likely have an impact in their preferred organisation scheme. As such, instead of providing a fully custom organisation and classification scheme, a personal digital repository needs to adopt and provide a baseline scheme that can be extended to support new content types, emerging properties for existing ones while at the same time is also able to encode within itself any additional organisation overlays that the repository owner chooses to create. Once again, it might be tempting to start from the ground up, yet there is already an organisation scheme that can be extended to accommodate the needs of a personal digital repository.

4.1 An Ontology For A Personal Digital Repository

The CIDOC/CRM model [30] can serve as the basis for an ontology for a personal digital repository. While the base model was developed to promote information interoperability (by offering a "*lingua-franca*") between memory institutions (for instance museums or archives) that use different metadata standards, it possesses a number of traits that can make its adaptation to personal scenarios desirable. Its intended scope includes the capture of the contextual information, including temporal information, that surrounds individual content pieces, a trait that is in alignment with the goals of a personal digital repository. The model itself was also intended to be both extensible and simplifiable. Extensible since it makes provisions to allow for additions that better represent the content that is going to be described and simplifiable since it recognises that not all of the classes, properties and relationships it defines might be useful for all scenarios, thus allowing for a subset of its entities, seen in Figure 4.1, to be used as a basis for other models and extensions while still maintaining compatibility with the full model. These twin traits can become important in the heterogeneous and dynamic environment of personal scenarios, where content types can gain properties over time, or new forms of content can appear and need to be integrated into the existing shared context. It also includes a type system that, through extension, serves as an entry point to the creation of alternative classification and organisation schemes, a property that is desirable in order to give the owners of a personal digital repository the possibility of creating a truly personal (even if only meaningful for themselves) organisation scheme. The type system can also be used as an alternative way to extend the CIDOC/CRM model, by introducing types from a controlled vocabulary instead of adding entities to it. For this particular scenario, the chosen route was to extend the model with additional properties and entities. This choice allows the introduction of entities when needed (for instance when the definition of the closest existing entities that could be type-casted restricts its scope to physical items) while mainly liberating the type system to be used by the repository owner as he sees fit to create his parallel classification scheme.

On the other hand, the CIDOC/CRM model's scope is also clearly aimed at the creation of "*scientific documentation*" which means that the depth and precision of the information created by this model should be enough to qualify for academic research. Even with the adoption of a simplified version of the model, it is likely that this organisation scheme can become overly detailed for some personal scenarios. The model is also event-oriented, which means that there is always an event mediating what might appear to be direct relations between entities. While this contributes to the incorporation of the temporal dimension (if available) it also introduces long paths to describe seemingly common relationships, increasing the complexity of the description. This issue is addressed in the model with the introduction of "shortcuts", which are properties that can be used to directly connect entities without defining a complete event chain. It should be noted that going through the event chain will also imply the existence of the appropriate "shortcut" property instance, yet two entities instances linked only by a "shortcut property" instance do not necessarily imply the existence of the event chain. In the personal digital repository scenario this can be explored to control the detail level according to what is known about content and its context, with more detail (and thus the use of longer event based chains) being assigned to personally produced content, and "shortcut" properties by themselves for when less details are known or to procured content.

To better support personal (and digital) scenarios, there is also the need to extend some parts of the CIDOC/CRM model. These extensions aim to introduce additional concepts and

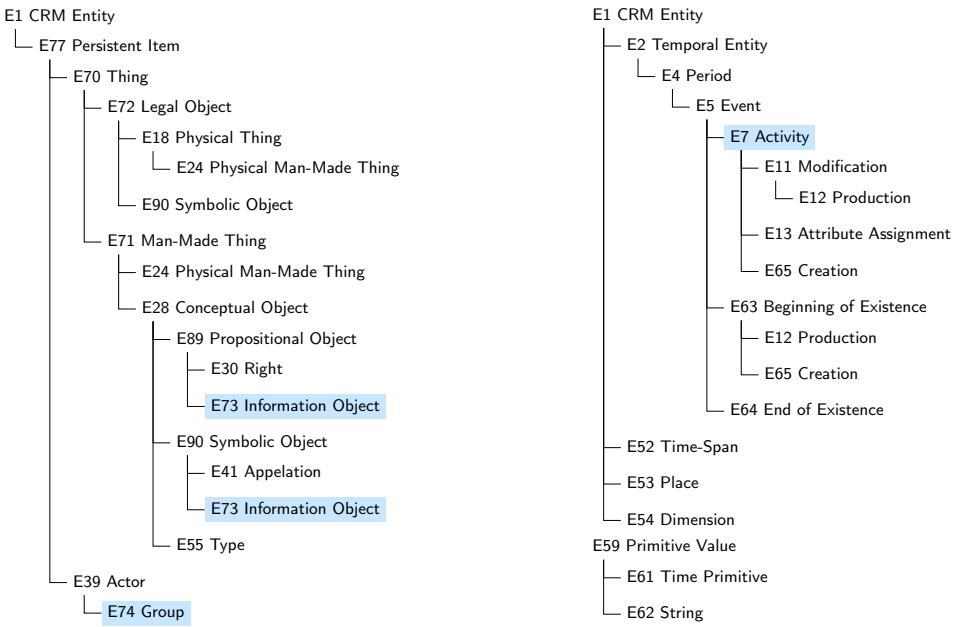


Figure 4.1: Minimum CIDOC/CRM subset

properties that can be used to describe digital content in the environment of a personal digital repository, but also to encode some of the repository's functional traits within the organisation scheme itself. The end objective of the introduced extensions is to be able to answer some of following questions regarding a given piece of content placed on the repository and their surrounding context:

- When was the content created or acquired?
- If not created by the repository owner where has the content came from?
- Who was in position to influence the creation or acquisition of the content?
- Where was the repository owner when the content was created or acquired?
- What other content pieces may be related to the content?
- How is the personal digital repository managing the content?

The first four questions serve to establish the digital content's context, be it temporal, social or spatial, with the fifth one tapping into the context to spur further navigation or exploration. The last question serve as control, by placing the focus on the repository itself, encompassing information encoded regarding the activities performed by the repository over the described content. It should be noted that for a given piece of content not all of these questions may actually have an answer, be it because of missing information that is not available in digital format to be gathered to the personal digital repository (for instance something that only the repository owner knows) or because of external real world events that have a direct influence in the content but that have no representation in the personal digital repository.

The primary extension point for digital content within the CIDOC/CRM model is the E73 Information Object entity. In the model's definition, that can be applied to digital content, this class represents immaterial items that possess a recognisable structure and should be documented as a single unit and whose instances do not depend of a physical carrier, and can exist in one or more carriers simultaneously. Choosing to model digital content as E73 Information Object brings with it issues with collections in the model. The entity that represents collections in the CIDOC/CRM model E78 Collection scope's definition states that collections are aggregations of E18 Physical Thing which the model previously establishes as being disjointed from one of the E73 Information Object's parent classes. Thus, collections of digital content can not be modelled using the provided class and instead will need to be modelled by a parallel entity. In the same way that physical collections are modelled as being extensions to the objects they contain (i.e. E18 Physical Thing is (through a path) a superclass of E78 Collection), digital collections should then be modelled as being extensions of the base class that will represent digital content. This approach ensures that a digital collection can also be treated as content in its own right. As it would be expected, the model does not have classes to represent the personal digital repository itself, nor the repository owner. While it is possible to argue that these can be modelled only with the introduction of additional properties, promoting them to first class entities enables their participation in event chains. For representing the repository owner the primary extension point within the model is the E21 Person entity, that is intended to represent real persons. Representing the personal digital repository itself requires more thought. Traditional repositories are often modelled as independent organisations with recognised statutes and norms that allow them to act as legal entities, which in the CIDOC/CRM model would mean that they would be represented by an instance of the E40 Legal Body entity. Yet a personal digital repository does not possess such characteristics, with its legal representation being left to the repository owner, and the only actions it can take are on the behalf of its owner and only over content. A personal digital repository is thus more akin to a human construction or artefact, albeit one that is more conceptual than physical in its nature, to the point that its owner may not know the specific physical vessel (or vessels) where it resides (for instance in the case of repositories provided as services). Given these traits, the most appropriate extension point within the model appears to be the E71 Man Made Thing that is intended to represent man made entities that may or may not have a physical form and that should be documented as a single entity. In addition to content, the model's support for relationships between individuals also needs to be addressed and extended. The model includes provisions to establish basic child-parent relationships, though these should only be interpreted in the biological sense, through event paths and "shortcut" properties. Families, in the social meaning of the term are hinted that should be described as instances of E74 Group. In spite of the previously mentioned provisions, this is arguably an area where the base CIDOC/CRM model needs to be expanded in order to provide support for ethnographic collections, or as is the case for personal digital collections. Taking what is already provided into account, social relationships can be described by extending the E74 Group concept, with the caveat that there is the need to add additional properties to describe the roles of the participants. By materialising individual instances of social relations it becomes possible to track their evolution through time, reinforcing the notion that in spite of being conceptually the same, two social relations of the same type can be very different from one another. Social relations are also noteworthy since they expose the underlying bias of a personal digital repository, as they are always described from someone's point of view. While simplistic models might assume that social

relations are reciprocal (for instance, friendship) that it not always the case, as one of the participants might perceive it in a way very different from the others (or even be unaware of its existence). The social relation description bias should be acknowledged, both by stating here that all relations, regardless of its kind or participants are seen from the perspective of the repository owner, as well as by defining properties that make explicit the role taken by the participant in the relation and that (if needed) are left without a formal inverse property. An additional constraint comes from the fact that the description of any relation is going to be limited by the available data, which means that there might be a gap between what gathered content indicates (for instance that two individuals are unrelated) and what might actually be happening. This both reinforce the notion that a personal digital repository is a biased environment and serves as additional justification for the need to have an open world model, where the conclusion reached for lack of data is 'unknown' instead of a hard 'no'. A direct consequence of the previously mentioned issue is that not all relations will have an associated event path, having to rely instead solely on weaker "shortcut properties". Finally, the model already provides entities that allows the description of spatial information (with the use of the E53 Place and the derivatives of the E44 Place Appellation entity) and to deal with contact information (with the use of the E51 Contact Point), though for this last case specific entities will be defined to represent common types of contacts and to specify that they too are digital objects. To better help the visualisation of how the forthcoming proposed entities are organised relative to each other and to existing CIDOC/CRM entities, the proposed entities has been placed in the hierarchical reference lists seen in Figure 4.2. In these list entities that came from the CIDOC/CRM model are highlighted, since they served as the extension points from which the proposed entities were derived. Since every proposed entity is an extension of an existing entity from the CIDOC/CRM, it follows that they can all be described, albeit in more generic terms and with some loss of information, by solely using the CIDOC/CRM model. This trait can be used to achieve interoperability with other systems as long as the vanilla CIDOC/CRM model is used as an intermediary.

It should be noted that there is no reliable way to predict how content will evolve over time, nor how the interests of the repository owner will change. This means that in order to support the natural evolution of both, these extensions proposed to the CIDOC/CRM model to support personal scenarios are not the end, but the beginning. While the repository extension may provide some ready to use extensions for well know content types, repository modules can and should bring with them extensions to the vocabulary and properties provided by both the CIDOC/CRM model and the personal digital repository extensions. It is the only route through which a personal digital repository can evolve over time to accommodate new content types and their properties.

4.1.1 Extensions to CIDOC entities

The following subsection contains the descriptions of the additional entities added to the CIDOC/CRM model in order to bolster support for personal digital repositories, both by complementing existing CIDOC/CRM entities with digital counterparts, events and concepts unique to the personal scenario as well as by providing some entities to represent common content types that are likely to appear as the gathered content. It follows the same structure and conventions as the definitions in the CIDOC/CRM reference document, with the notable difference that entity names are prefixed with "EPDR" (which stands for Entity - Personal Digital Repository (EPDR)) instead of simply "E" (which stands for entity) to distinguish

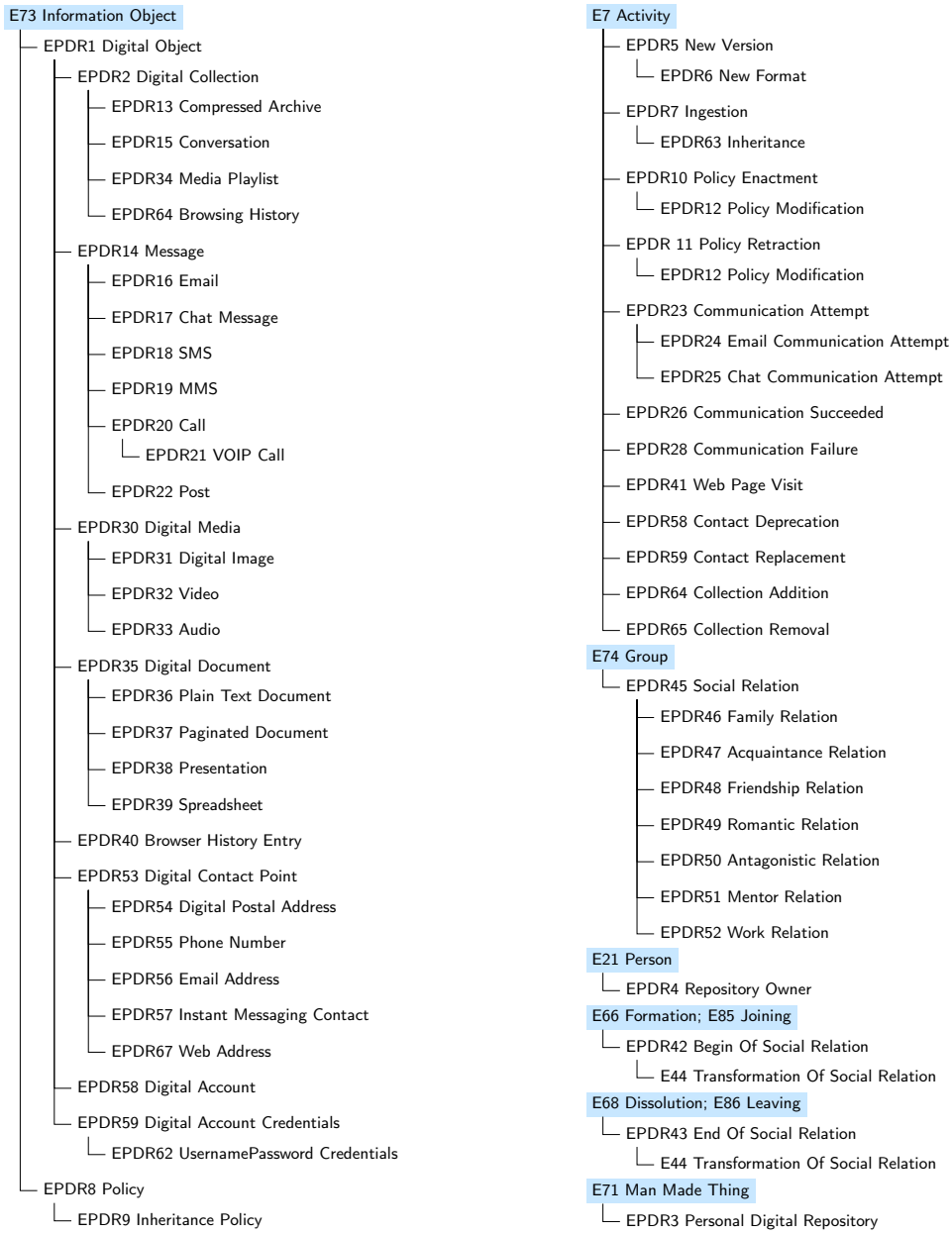


Figure 4.2: Hierarchical reference list for the proposed entities

them from those defined in the model proper.

EPDR1 Digital Object

Subclass of E73 Information Object

Superclass of EPDR2 Digital Collection, EPDR14 Message, EPDR30 Digital Media, EPDR35 Digital Document, EPDR40 Browser History Entry, EPDR53 Digital Contact Point, EPDR58 Digital Account, EPDR59 Digital Account Credentials

Scope note This entity forms the basis for gathered content in a personal digital repository. A digital object is a man made artefact that exist entirely in a digital format. It encompasses both traditional file based content as well as information pieces transmitted through digital means and to which end users usually do not associate a contained representation. In a personal digital repository scenario it is assumed that the owner of a digital object is also the owner of the personal digital repository it currently resides, though it may not have been the only owner or even the sole author of the digital object. A digital object can have any number of authors or contributors, though only one can be considered to be the canonical creator (in the sense that it brought the digital object into existence by creating the initial carrier).

Example

- A downloaded pdf file
- A sent text message
- A browser history entry

Properties

- PPDR7 is new version of (is previous version of): EPDR1 Digital Object
- PPDR10 has current or former digital owner (is former or current digital owner of): E39 Actor
- PPDR11 has current digital owner (is current digital owner of): EPDR4 Repository Owner
- PPDR13 was inherited by (inherited): E39 Actor
- PPDR20 has current policy (currently applies to): EPDR8 Policy
- PPDR21 has former policy (formerly applied to): EPDR8 Policy

EPDR2 Digital Collection

Subclass of EPDR1 Digital Object

Superclass of EPDR13 Compressed Archive, EPDR15 Conversation, EPDR34 Media Playlist, EPDR64 Browsing History

Scope note This entity represents an aggregation of EPDR1 Digital Object that have been grouped together. Their aggregation might have come to past for functional reasons, as a mean to reinforce the connection between the EPDR1 Digital Object that comprise the collection or simply as an organisational aid created by the repository owner. It should

be noted that a EPDR2 Digital Collection collection is by its own right also a piece of content, and thus can be manipulated as a single entity, though its disappearance by any means imply also the disappearance of the its composing EPDR1 Digital Object.

Example

- A file system folder
- All photos from a field trip
- A compressed archive file
- A conversation (i.e a sequence of related messages) between two individuals

EPDR3 Personal Digital Repository

Subclass of E71 Man Made Thing

Scope note This entity represents the incorporeal concept of a personal digital repository. While a particular instance may represent the current personal digital repository which is property of the repository owner, gathered content may indicate that other individuals also possess their owner personal digital repositories about which some properties may be known, such as their location (i.e. how to access them).

Example

- John's Personal Digital Repository

Properties

- PPDR5 has repository owner (owns personal digital repository): E39 Actor

EPDR4 Repository Owner

Subclass of E21 Person

Scope note This entity represents the owner of the current personal digital repository. All content gathered in the current personal digital repository is assumed to be owned (though not necessarily created) by the repository owner.

Example

- John Doe, the owner of John's Personal Digital Repository

EPDR5 New Version

Subclass of E7 Activity

Scope note This entity represents an event by which a new version of an existing EPDR1 Digital Object is produced. New versions of an existing EPDR1 Digital Object can be created due to a myriad of reasons, ranging from being the result of modifications done to the original EPDR1 Digital Object by the repository owner (or other contributors) between collection cycles to a change of container format. Their common trait is that they produced a noticeable change that made the resulting EPDR1 Digital Object distinct from the original one, while remaining conceptually linked. Though linked to an existing EPDR1 Digital Object, the resulting EPDR1 Digital Object has its own set

of properties that can be substantially different from those of the original, is distinct from the original one and can even exist simultaneously with its previous version, thus allowing the repository owner to track how a conceptual object changed through time. On the other hand, this event allows the establishment of a temporal relation between the existing EPDR1 Digital Object and the resulting EPDR1 Digital Object as well as causal relation (as without the original EPDR1 Digital Object, the new version could not have existed).

Example

- The submission of a modified version of a file by a content collection tool
- The conversion of the underlying bytestream of a EPDR1 Digital Object to another format by the personal digital repository

Properties

- PPDR6 used as source (was used as source by): EPDR1 Digital Object
- PPDR81 resulted in (was the result of): EPDR1 Digital Object

EPDR6 New Format

Subclass of EPDR5 New Version

Scope note This entity represents an event by which an existing EPDR1 Digital Object is converted to a different representation format. There is no warranty that the new representation format is appropriate for the actual content of the original EPDR1 Digital Object, nor that the transformation will be able to translate all the features of the original EPDR1 Digital Object.

Example

- The conversion of the underlying bytestream of a EPDR1 Digital Object to another format by the personal digital repository

EPDR7 Ingestion

Subclass of E7 Activity

Scope note The entity represents the event that lead to the acquisition of a EPDR1 Digital Object by the current personal digital repository. In a personal scenario it means that the personal digital repository has become aware of the existence of a given content piece, creating an internal representation of it in the form of a EPDR1 Digital Object. It does not mean that the content has been transferred to the personal digital repository (since it can reside outside of it). When an EPDR1 Digital Object is ingested into the current personal digital repository it also means that as far as the repository is concerned the repository owner becomes the effective owner of said digital object, though this does not necessarily mean that there had been a precious change of ownership, be it legal or otherwise. Thus, this event can serve as a rough approximation for the acquisition of ownership over the EPDR1 Digital Object.

Example

- The collection of a text message by a content collection tool

- The submission of a file to the personal digital repository by the repository owner

Properties

- PPDR8 acquired digital object (was acquired in): EPDR1 Digital Object
- PPDR9 granted current or former ownership to (had current or former ownership by): E39 Actor
- PPDR14 acquired to repository (content increased by): EPDR3 Personal Digital Repository

EPDR8 Policy

Subclass of E73 Information Object

Scope note This entity represents and abstract policy that is associated with a EPDR1 Digital Object on the current personal digital repository. Policies are kept associated with the digital object in order to track their evolution through time, with this being an description what the effect of said policy would be (as its actual definition and implementation is left for the personal digital repository itself).

Example

- The privacy settings of a given content piece
- The *Allowed Inheritance List* of a given content piece

EPDR9 Inheritance Policy

Subclass of EPDR8 Policy

Scope note This entity represents an inheritance policy that is associated with a EPDR1 Digital Object on the current personal digital repository. An EPDR1 Digital Object can have multiple associated inheritance policies, both active and historical though only the active ones are expected to be acted upon.

Example

- The *Allowed Inheritance List* of a given content piece

EPDR10 Policy Enactment

Subclass of E7 Activity

Scope note This entity represents the event that lead to a EPDR8 Policy becoming active, and thus affecting a given EPDR1 Digital Object.

Example

- Setting a given digital object as *Not Inheritable*

Properties

- PPDR15 enacted (enacted by): EPDR8 Policy
- PPDR16 enacted over (affected by enactment): EPDR1 Digital Object

EPDR 11 Policy Retraction

Subclass of E7 Activity

Scope note This entity represents the event that lead to a EPDR8 Policy becoming inactive, and thus stop being applied to a given EPDR1 Digital Object.

Example

- Removing the *Not Inheritable* policy from a picture

Properties

- PPDR17 retracted (retracted by): EPDR8 Policy
- PPDR18 retracted over (affected by retraction): EPDR1 Digital Object

EPDR12 Policy Modification

Subclass of EPDR10 Policy Enactment, EPDR11 Policy Retraction

Scope note This entity represents the event that lead to a change in a EPDR8 Policy applied to a given EPDR1 Digital Object. The change entails the retraction of a previously applied EPDR8 Policy and the enactment of its replacement. If the replacement policy to be enacted is the default policy, then the event is better described as being a EPDR11 Policy Retraction instead of a EPDR12 Policy Modification. It should be noted that it is the personal digital repository's responsibility to distinguish between EPDR11 Policy Retraction and EPDR12 Policy Modification when recording the EPDR1 Digital Object event history.

Example

- Setting a given digital object's *Allowed Inheritance List* when it was previously marked as *Not Inheritable*.

EPDR13 Compressed Archive

Subclass of EPDR2 Digital Collection

Scope note This entity represents a compressed archive. Compressed Archive files serve as containers for other EPDR1 Digital Object, thus behaving like an impromptu digital collection. While a digital collection is mainly conceptual and does not have the need for a specific carrier vessel, Compressed Archives are limited by the need to have a carrier vessel (usually a file). The carrier itself may be monolithic or divided in several pieces, all of which are needed to retrieve the compressed archives' content.

Example

- A zip file

EPDR14 Message

Subclass of EPDR1 Digital Object

Superclass of EPDR16 Email, EPDR17 Chat Message, EPDR18 SMS, EPDR19 MMS, EPDR20 Call, EPDR22 Post

Scope note This entity represents a generic communication unit exchanged between actors that are unable to communicate directly and thus choose to use a digital surrogate in order to exchange information. To achieve this, the surrogates are sent to known contact points that can be used by the actors to retrieve the communications.

Example

- A text message from John to Mary
- An email sent by an acquaintance

Properties

- PPDR25 sent to (was used to receive): EPDR53 Digital Contact Point
- PPDR26 sent from (was used to send): EPDR53 Digital Contact Point
- PPDR27 has next message (has previous message): EPDR14 Message

EPDR15 Conversation

Subclass of EPDR2 Digital Collection

Scope note This entity represents an ordered collection whose contents are composed only by EPDR14 Message entities. Unlike EPDR2 Digital Collection, where the order by which the contents are added or presented may not be relevant, a EPDR15 Conversation imposes order (determined for instance by temporal criteria or sequence numbers) upon its contents to preserve the original flow of the communication.

Example

- An email thread
- The log of a text chat between John and Mary

EPDR16 Email

Subclass of EPDR14 Message

Scope note This entity represents a communication specifically exchanged by using electronic mail (email) comprised of textual content and possibly attachments.

Example

- An email exchanged between the repository owner and an acquaintance.

EPDR17 Chat Message

Subclass of EPDR14 Message

Scope note This entity represents a text message exchanged using an instant messaging or chat communication protocol.

Example

- An Internet Relay Chat (IRC) message to a group chat.
- A private instant message exchanged between the repository owner and Mary.

EPDR18 SMS

Subclass of EPDR14 Message

Scope note This entity represents a text message exchanged using the Short Message Service (SMS), usually in a mobile environment. These messages are textual in nature and do not contain any attachments.

Example

- An SMS received by the repository owner in his mobile phone

EPDR19 MMS

Subclass of EPDR14 Message

Scope note This entity represents a message exchanged using the Multimedia Messaging Service (MMS), usually in a mobile environment. The content of this message type can include pictures, sounds or videos as attachments in addition to their textual content.

Example

- An MMS sent by the repository owner form his mobile phone

EPDR20 Call

Subclass of EPDR14 Message

Scope note This entity represents a voice communication between actors. This can encompass communications made with traditional telephony systems (be them fixed or mobile) or alternative Voice over IP (VoIP) protocols.

Example

- A VoIP call between the repository owner and a family member
- A telephone call between the repository owner and a acquaintance

EPDR21 VOIP Call

Subclass of EPDR20 Call

Scope note This entity represents a voice communication specifically made using a VoIP protocol through means other than the fixed land lines or the mobile voice service from a mobile carrier.

Example

- A VoIP call between the repository owner and a family member.

EPDR22 Post

Subclass of EPDR14 Message

Scope note This entity represents a communication posted specifically in an online community such as message boards or social media services. While other message types are primarily concerned with actor to actor communication, EPDR22 Post are open ended. As such they do not necessarily target a specific actor with whom to initiate a conversation but instead invite the participation of multiple actors. If placed upon public boards or services they may be answered by actors unknown to the post creator.

Example

- A post on an online board
- A social network status update

EPDR23 Communication Attempt

Subclass of E7 Activity

Scope note This entity represents the event that marks the point in time that actors attempted to communicate with each other, with the direction of the communication being determined by the properties associated with its instances. This compact representation allows the collapse onto a single event of multiple events that are perceived in a different way depending on which side of communication one is (i.e. sending or receiving a message), as those multiple events are essentially mirror images of each other given a different interpretation according to the direction of communication as seen by an observer. This event can not be used to make assertions regarding if communication attempt was successful, yet it can be used to provide temporal reasoning and establishing temporal order regarding messages (i.e. to establish a time before a communication attempt was made).

Example

- Sending a chat message to a friend
- Initiating a call

Properties

- PPDR22 communicated (was communicated by): EPDR14 Message

- PPDR23 communicated with (received communication in): EPDR53 Digital Contact Point
- PPDR24 communication from (sent communication in): EPDR53 Digital Contact Point

EPDR24 Email Communication Attempt

Subclass of EPDR23 Communication Attempt

Scope note This entity represents the event that marks the point in time that actors attempted to communicate with each other, with the direction of the communication being determined by the properties associated with its instances. The contact attempt was made specifically using email messages, and thus this event should not be applied to any other message entities.

Example

- Sending an email to an acquaintance

EPDR25 Chat Communication Attempt

Subclass of EPDR23 Communication Attempt

Scope note This entity represents the event that marks the point in time that actors attempted to communicate with each other, with the direction of the communication being determined by the properties associated with its instances. The contact attempt was made specifically using online chat messages, and thus this event should not be applied to any other message entities.

Example

- Sending an email to an acquaintance

EPDR26 Communication Succeeded

Subclass of E7 Activity

Scope note This entity represents a follow-up event to the EPDR23 Communication Attempt, marking a point in time where a given communication attempt was successful. While primarily aimed at communication attempts where the repository owner is the intended recipient, it can be applied to communications between any set of actors. It should be noted that this event may not always be present (for instance since the collection is unable to determine when it happened). Its presence only ensures that the intended recipient was contacted successfully and has viewed, heard or otherwise interacted with the message passed on to him, though it makes no assertions regarding if it was able to understand its contents.

Example

- Opening a received SMS for the first time.
- Opening a received email for the first time.
- Taking a call from an acquaintance.

EPDR28 Communication Failure

Subclass of E7 Activity

Scope note This entity represents a follow-up event to the EPDR23 Communication Attempt, marking a point in time where a given communication attempt was not successful. While primarily aimed at communication attempts where the repository owner is the intended recipient, it can be applied to communications between any set of actors. It should be noted that this event may not always be present (for instance since the collection is unable to determine when it happened). Its presence only ensures that the intended recipient was not contacted successfully and thus has not viewed, heard or otherwise interacted with the message passed on to him.

Example

- The repository owner missing a call from an acquaintance on his mobile phone.

EPDR30 Digital Media

Subclass of EPDR1 Digital Object

Superclass of EPDR31 Digital Image, EPDR32 Video, EPDR33 Audio

Scope note This entity represents a digital object whose purpose is to convey a representation of media objects in a digital format, regardless of their underlying carrier format.

Example

- A music by the repository owner's favourite artist
- A photography of the last vacation of the repository owner.
- A video of the last vacation of the repository owner.

EPDR31 Digital Image

Subclass of EPDR30 Digital Media, E38 Image

Scope note This entity represents a digital object whose purpose is to convey a primarily static representation of visual media in a digital format.

Example

- A photography of the last vacation of the repository owner.
- A screen shot of a web page's appearance at the time of visit.
- An animation in the Graphics Interchange Format (GIF) from a web page.

EPDR32 Video

Subclass of EPDR30 Digital Media

Scope note This entity represents a digital object whose purpose is to convey a moving representation of visual media in a digital format.

Example

- A video of the last vacation of the repository owner.
- A video stream from an online media provider.

EPDR33 Audio

Subclass of EPDR30 Digital Media

Scope note This entity represents a digital object whose purpose is to convey a representation of audio media in a digital format.

Example

- A music by the repository owner's favourite artist

EPDR34 Media Playlist

Subclass of EPDR2 Digital Collection

Scope note This entity represents an ordered collection whose contents are composed only by EPDR30 Digital Media entities. Unlike an EPDR2 Digital Collection, were the order by which contents are presented may not be relevant, a EPDR24 Media Playlist imposes order upon its contents in order to represent a given aesthetic choice. The order can be defined by the repository owner or by a third party when the content was released.

Example

- A digital audio album
- A series of videos that cover the same subject in an online media sharing platform

EPDR35 Digital Document

Subclass of EPDR1 Digital Object

Superclass of EPDR36 Plain Text Document, EPDR37 Paginated Document, EPDR38 Presentation, EPDR39 Spreadsheet

Scope note This entity represents a digital object whose purpose is to convey information in a structured manner, be that structure be intrinsic to the representation format or to the content itself.

Example

- A Portable Document Format (PDF) file
- A computer program source file
- A spreadsheet file

EPDR36 Plain Text Document

Subclass of EPDR35 Digital Document

Scope note This entity represents a EPDR35 Digital Document that is specifically tailored for the exchange of content structured and encoded in such a way that it can be read (though not necessarily understandable) by humans without the help of the applications that created it, or applications specifically designed for its interpretation other than generic applications capable of displaying textual content. As such the content contained within instances of this entity is expected to be composed only by text, with its structure being encoded within the content itself.

Example

- A plain text file
- A computer program source file
- A XML file
- A JSON file

EPDR37 Paginated Document

Subclass of EPDR35 Digital Document

Scope note This entity represents a EPDR35 Digital Document that is specifically tailored for the exchange of content structured in such a way that it becomes human readable (though not necessarily understandable) after it is decoded by applications who can understand the underlying carrier format. The decoded form is able to convey the notion that its content is primarily divided and structured according to the traditional notion of pages. Upon decoding, content contained within instances of this entity is expected to be primarily textual in nature, but can also include embedded in itself other media formats to further reinforce the content's subject or to serve as composition aides.

Example

- A PDF file
- A document in Microsoft's Word format

EPDR38 Presentation

Subclass of EPDR35 Digital Document

Scope note This entity represents a EPDR35 Digital Document that is specifically tailored to serve as support materiel for presentations. Its content is structured in such a way that it becomes readable (though not necessarily understandable) by humans after it is decoded by applications who can understand the underlying carrier format. The decoded form of these entities may encompass multiple media types, depending on the underlying carrier support and the needs of the subject matter being presented.

Example

- A Beamer generated PDF file
- A presentation in Apple's Keynote file format

EPDR39 Spreadsheet

Subclass of EPDR35 Digital Document

Scope note This entity represents a EPDR35 Digital Document that is specifically tailored to serve as container for content in tabular formats. Its content is structured in such a way that it becomes readable (though not necessarily understandable) by humans after it is decoded by applications who can understand the underlying carrier format. The decoded form of these entities can contain instructions on how to calculate values from other parts of the content, as well as other media types to convey additional representation of the data provided by the entity.

Example

- A spreadsheet in Libreoffice's Calc file format

EPDR40 Browser History Entry

Subclass of EPDR1 Digital Object

Scope note This entity represents am entry in the history of a web browser, made by visiting a web page at a given point in time, and function as a reflection of how the visited page was at the time of the visit. They serve as aggregation point for information collected regarding the page itself, be it provided by the browser history engine or gleaned from the page itself, for instance, collected from Resource Description Framework in Attributes (RDFa) statements about the page at visit time. Visiting the same page repeatedly will yield distinct entries that can possibly carry with them changes in the collected information (for instance multiple visits to the main page of a news site, where its contents change over time).

Example

- The entry on the Firefox browser history left by the visit to www.google.com on 01/01/2015

Properties

- PPDR29 has web page address (is web page address of): EPDR65 Web Address

EPDR41 Web Page Visit

Subclass of E7 Activity

Scope note This entity represents a visit to a given web page that results in the creation of a EPDR40 Browser History Entry. The visit can take place from any device and web browser that is nominally under the control of the repository owner, though this does not ensure that it has been initiated by the repository owner himself. As an event it can be used to perform temporal reasoning about the visits (i.e. establish a time before the page was visited, or after the visit has occurred).

Example

- The visit to www.google.com on 13:24:51 01/01/2015

Properties

- PPDR28 created entry (was created in): EPDR40 Browser History Entry
- PPDR29 visited (was visited in): EPDR65 Web Address

EPDR42 Begin Of Social Relation

Subclass of E66 Formation, E85 Joining

Superclass of E44 Transformation Of Social Relation

Scope note This entity represents the event by which a social relation between actors has formed, represented by the creation and immediate joining of a group that represents said social relation by its participants. As an event it can be used to provide temporal reasoning about the social relation at hand (i.e. to establish a time before it existed and one after it became effective), though it can not be used to make any kind of assertion about who took the initiative to begin the social relationship, nor if all of its actors are willing participants in it.

Example

- The repository owner makes the acquaintance of another person
- The repository owner becomes to see another person as a friend

EPDR43 End Of Social Relation

Subclass of E68 Dissolution, E86 Leaving

Superclass of E41 Transformation Of Social Relation

Scope note This entity represents the event by which a social relation between actors has been terminated, represented by the abandonment and immediate dissolution of the group that represents said social relation by its participants. As an event it can be used to provide temporal reasoning about the social relation at hand (i.e. to establish a time after the end of the relation, or in conjunction with EPDR41 Begin Of Social Relation to determine and approximate time span for its duration), though it can not be used to make any kind of assertion regarding who took the initiative to terminate the social relation, nor if all participants left the relation willingly.

Example

- The repository owner stops seeing another person as a friend.
- A person stops being a co-worker of the repository owner.

EPDR44 Transformation Of Social Relation

Subclass of EPDR42 Begin Of Social Relation, EPDR43 End Of Social Relation

Scope note This entity represents the event where a given social relation suffered a such a modification that it effectively ceases to be applicable, while at the same time a new kind of social relation takes it place, effectively marking a boundary between both relations. This modification in the relation status is represented by the abandonment and dissolution of the group that represented the previous relation, and the creation and joining of the group that represents the new relation, with the roles taken by the involved relation being determined by the entity's properties. As an event it can be used to provide temporal reasoning regarding the involved social relations, establishing the temporal boundary between them.

Example

- The marriage of the repository owner with its significant other.
- A person stops being a co-worker of the repository owner but they remain friends.

EPDR45 Social Relation

Subclass of E74 Group

Superclass of EPDR46 Family Relation, EPDR47 Acquaintance Relation, EPDR48 Friendship Relation, EPDR49 Romantic Relation, EPDR50 Antagonistic Relation, EPDR51 Mentor Relation, EPDR52 Work Relation

Scope note This entity represents a generic social bond between actors. Though it assumes multiple participants, not all of them are required to be aware of the existence of the relation or agree on its nature. The nature and direction of the relation is defined by the properties associated with its instances, and the same participants can be in multiple relations at the same time. In a personal digital repository scenario, all relations are biased since they are represented from the perspective of the repository owner.

Example

- Two individuals who have just met for the first time, becoming acquaintances

EPDR46 Family Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond between actors that is primarily familiar in nature, as seen by the repository owner. Its nature refers to the social aspects of family, not the biological aspects of it. It can be used to represent both traditional family blood ties as well as social family ties, with the nature and direction of the relation defined by the properties associated with its instances.

Example

- The relation between John and his parents.

Properties

- PPDR47 has parent in (is parent in): E39 Actor
- PPDR52 has child in (is child in): E39 Actor
- PPDR56 has sibling in (is sibling in): E39 Actor
- PPDR61 has uncle/aunt in (is uncle/aunt in): E39 Actor
- PPDR65 has nephew/niece in (is nephew/niece in): E39 Actor
- PPDR68 has cousin in (is cousin in): E39 Actor
- PPDR70 has grandparent in (is grandparent in): E39 Actor
- PPDR73 has grandchild in (is grandchild in): E39 Actor

EPDR47 Acquaintance Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond between actors in its early stages of development, as seen by the repository owner. Participants in this relationship know about the existence of the others and interact with each other if needed. However they refrain from overly exposing themselves to other participants, limiting the trust placed on one another.

Example

- Two individuals who have just met for the first time.

Properties

- PPDR30 has acquaintance (is acquaintance in): E39 Actor

EPDR48 Friendship Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond stronger than EPDR47 Acquaintance Relation in which one of the participants (though not necessary all of them) trust in the others, making their opinion more valuable to them and overall increasing their influence. Higher degrees of trust can lead to the further exposure between the participants, that may become willing to reveal more information about themselves to the other participants without fear of negative reactions. The direction of the relation is defined by the properties associated with its instances.

Example

- The repository owner spent enough time with an acquaintance to starting considering her a friend.

Properties

- PPDR32 has friend in (sees as friends those in): E39 Actor
- PPDR33 has been considered friend in (seen as friend by some in): E39 Actor

EPDR49 Romantic Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond whose scope has transcended friendship and became a deeper emotional and/or physical relation punctuated by strong feelings between the participants. It is assumed that all participants in this relation type are aware of their part in the relation, making it effectively symmetrical (in this particular case asymmetrical relations should be considered a product of one the participant's mind, and thus be purely objects conceptual that express desires). This relation type can serve as the prelude to the formation of a family relationship, should the bond between the participants be strong enough, or to an antagonistic relation if its failure degrades the opinion of the participants in an significant way.

Example

- The relation between the repository owner and his significant other

Properties

- PPDR35 has romantic partner (is involved in): E39 Actor

EPDR50 Antagonistic Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond by which one of the participants expresses its dislike for the another, though the feeling is not necessarily mutual. The direction of the relation is defined by the properties associated with its instances.

Example

- The relation between the repository owner and his former significant other

Properties

- PPDR37 has been antagonised in (sees as antagonists those in): E39 Actor
- PPDR38 has been considered antagonist in (seen as antagonist by some in): E39 Actor

EPDR51 Mentor Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond by which one of the participants is seen as an experienced adviser or tutor that can guide or serve as a role model for the other participant in the relationship. The guiding received might be formal or informal and is not necessarily acknowledged as such by all participating parties. The direction of the relation is defined by the properties associated with its instances.

Example

- The relation between the repository owner and his teacher.

Properties

- PPDR40 has pupil in (sees as mentors those in): E39 Actor
- PPDR41 has been considered mentor in (seen as mentor by some in): E39 Actor
- PPDR42 has mentor in (sees as pupils those in): E39 Actor
- PPDR43 has been considered pupil in (seen as pupil by some in): E39 Actor

EPDR52 Work Relation

Subclass of EPDR45 Social Relation

Scope note This entity represents a social bond motivated by the sharing of a common working environment. It can be used to describe the relation between co-workers on the same level, or between hierarchical superiors and their subordinates, with the nature and direction of the relation being defined by the properties associated with its instances. This type of relation can be the underlying cause for the establishment of other relation types. For instance two co-workers who share the same workspace are also acquainted (EPDR47 Acquaintance Relation) with each other, and this acquaintance relation can evolve (EPDR44 Transformation Of Social Relation) over time to become one of the other relation types. These other relations that are spurred by a professional relation will often outlast the EPDR52 Work Relation, which has a tendency to be transient, with its end marked by changes in the work place or environment.

Example

- The relation between the repository owner and his co-workers

Properties

- PPDR46 has coworker in (is coworker in): E39 Actor

EPDR53 Digital Contact Point

Subclass of EPDR1 Digital Object, E51 Contact Point

Superclass of EPDR54 Digital Postal Address, EPDR55 Phone Number, EPDR56 Email Address, EPDR57 Instant Messaging Contact, EPDR67 Web Address

Scope note This entity represents an identifiers for the places or means that can be used to contact an actor. These identifiers can be provided as part of the services associated with digital accounts, or be representations of real world entities (such as a street name). While they are primarily identifiers, they are also digital objects, since the information they encode is often sought and exchanged by itself (i.e. individuals aim to exchange contact information) and much like traditional documents it can change overtime (i.e. new identifiers can arise that supersedes the previous ones).

Example

- The address 123, Fake Street
- The phone number +351 200 000 000
- The email address personalDigitalRepositoryOwner@example.com

EPDR54 Digital Postal Address

Subclass of EPDR53 Digital Contact Point, E45 Address

Scope note This entity represents a digital object specifically created to describe the identifier for physical places that can be used to contact an actor, such as postal addresses.

Example

- The address 123, Fake Street

EPDR55 Phone Number

Subclass of EPDR53 Digital Contact Point

Scope note This entity represents a digital object specifically created to describe the identifier for a phone number that can be used to contact an actor.

Example

- The phone number +351 200 000 000
- The mobile phone number +351 916 900 000

EPDR56 Email Address

Subclass of EPDR53 Digital Contact Point

Scope note This entity represents a digital object specifically created to describe the identifier for an email address that can be used to contact an actor.

Example

- An email address

EPDR57 Instant Messaging Contact

Subclass of EPDR53 Digital Contact Point

Scope note This entity represent a digital object specifically created to describe the identifier for an instant messaging account that can be used to contact an actor. This identifier should not be confused with the handle or screen name used by the actor when contacted through said instant messaging account.

Example

- The email address personalDigitalRepositoryOwner@example.com

EPDR58 Contact Deprecation

Subclass of E7 Activity

Scope note This entity represents the event by which a EPDR53 Digital Contact Point stops being a viable mean of contact for its associated actor. As an event it can be used to provide temporal reasoning about the contact at hand (i.e. to establish a time after which the contact point is no longer viable). This event makes no assertions regarding the continued existence of the contact point itself. For instance the contact point might have been reassigned to another (possibly unknown) individual or its holder may have lost the means to access it. In both cases the contact point to still exists yet it will no longer serve as a viable mean to contact its associated actor.

Example

- An acquaintance changes his address without informing the repository owner of a new contact address
- The repository owner removes a telephone number for an acquaintance

Properties

- PPDR77 deprecated contact (was deprecated in): EPDR53 Digital Contact Point

EPDR59 Contact Replacement

Subclass of E7 Activity

Scope note This entity represents the event by which a EPDR53 Digital Contact Point is replaced as a viable mean of contact by another, different EPDR53 Digital Contact Point. As an event it can be used to provide temporal reasoning about the contacts at hand (i.e. to establish a temporal boundary over which the new contact superseded the previous one). Like the EPDR58 Contact Deprecation this event makes no assertions regarding the continued existence of the replaced contact point, with same caveats applying.

Example

- An acquaintance changes his address but informs the repository owner of a new contact address

Properties

- PPDR78 has replacement contact (became replacement contact in): EPDR53 Digital Contact Point

EPDR60 Digital Account

Subclass of EPDR1 Digital Object

Scope note This entity represents the arrangements by which a specific actor is given access to specific online services, a specific device or a specific application. Access to the resource is mediated by a set of credentials, typically a combination of username and password, though other options can be available.

Example

- An email account
- A social media account

Properties

- PPDR80 has current or former credentials (is current or former credential for): EPDR61 Digital Account Credentials

EPDR61 Digital Account Credentials

Subclass of EPDR1 Digital Object

Scope note This entity represents the credentials used to access a specific digital account. Possession of correct set of credentials means effective control over the account to which they are associated, regardless of who actually owns the account.

Example

- A username/password combination that grants access to an email account

EPDR62 UsernamePassword Credentials

Subclass of EPDR61 Digital Account Credentials

Scope note This entity represents the credentials used to access a specific digital account, specifically the commonly used username and password combination.

Example

- A username/password combination that grants access to an email account

EPDR63 Inheritance

Subclass of EPDR7 Ingestion

Scope note This entity represents the event by which digital objects are passed through from a given actor to another after the demise of the first. This can be a recurring event (i.e. a given digital object can have been passed along multiple actors), forming an inheritance chain. As an event it can be used for temporal reasoning about the digital objects (i.e. by establishing a time before the digital object was placed in the digital repository of the inheritor).

Example

- The ingestion of a digital object due to an inheritance policy triggered in another personal digital repository

Properties

- PPDR12 bequeathed as heirloom (was bequeathed as heirloom in): EPDR1 Digital Object

EPDR64 Collection Addition

Subclass of E7 Activity

Scope note This entity represents the event by which an EPDR2 Collection was augmented by the addition of another EPDR1 Digital Object. As an event it can be used for temporal reasoning about the collection and the participating digital objects (i.e. by establishing a time before the digital object belong to the collection).

Example

- The repository owner adds a digital object to a collection
- A new entry in an online chat conversation

Properties

- PPDR1 added to digital collection (was expanded by): EPDR2 Digital Collection
- PPDR3 added digital object (was made part of by): EPDR1 Digital Object

EPDR65 Collection Removal

Subclass of E7 Activity

Scope note This entity represents the event by which an EPDR2 Collection was diminished by the removal of one or more of its EPDR1 Digital Object. Ad an event it can be used for temporal reasoning about the collection and the participating digital objects (i.e. by establishing a time when the digital object still belonged to the collection).

Example

- The repository owner removes a digital object from a collection

Properties

- PPDR2 removed from digital collection (was removed by): EPDR2 Digital Collection
- PPDR4 removed digital object (removed as part by): EPDR1 Digital Object

EPDR66 Browsing History

Subclass of EPDR2 Digital Collection

Scope note This entity represents a collection whose contents are composed only by EPDR40 Browser History Entry entities. Unlike EPDR2 Digital Collection, were the order by which contents are presented may not be relevant, a EPDR66 Browsing History imposes order upon its contents a temporal order based upon the entry's visit date.

Example

- The collected entries of pages visited in the firefox browser of the repository owner.

EPDR67 Web Address

Subclass of E51 Contact Point

Scope note This entity represents a contact point identified by a uniform resource locator (URL), commonly known as a web address. They serve the dual role of being contact points for the actors that provide the machine accessed through them, as well as identifiers for specific resources that are contained within said machines, and whose access and retrieval may be the ultimate goal of the actor who accesses them.

Example

- http://www.example.com

4.1.2 Extensions to CIDOC properties

The following subsection contains the descriptions of the additional properties added to the CIDOC/CRM model in order to bolster support for personal digital repositories. It follows the same structure and conventions as the definitions in the CIDOC/CRM reference document, with the noble exceptions that properties names are prefixed with "PPDR" (which stands for Property - Personal Digital Repository) instead of simply "P" (which stands for property) to distinguish them from those defined in the model proper, and the avoidance of properties of properties, representing them instead as sub properties when appropriate. It should be noted that there properties that are already provided by the CIDOC/CRM model and which are not redefined here. For instance, the P106 is composed of (forms part of) property can be used as a shortcut property to indicate the composition of digital collections.

PPDR1 added to digital collection (was expanded by)

Domain EPDR64 Collection Addition

Range EPDR2 Digital Collection

Quantification many to many, necessary

Scope Note This property relates the EPDR62 Collection Addition event with the EPDR2 Digital Collection to which an digital object had been added to.

Examples

- The vacation photos collection (EPDR34 Media Playlist) *was expanded by* adding (EPDR64 Collection Addition) a new photo.

PPDR2 removed from digital collection (was reduced by)

Domain EPDR65 Collection Removal

Range EPDR2 Digital Collection

Quantification many to many, necessary

Scope Note This property relates the EPDR65 Collection Removal event with the EPDR2 Digital Collection in which an digital object had been removed from it.

Examples

- The vacation photos collection (EPDR34 Media Playlist) *was reduced by* removing (EPDR65 Collection Removal) the cat's photo.

PPDR3 added digital object (was made part of collection by)

Domain EPDR64 Collection Addition

Range EPDR1 Digital Object

Quantification many to many, necessary

Scope Note This property relates the EPDR64 Collection Addition event with the EPDR1 Digital Object which was added to a digital collection

Examples

- The airport photo (EPDR31 Digital Image) *was made part of collection by* adding (EPDR64 Collection Addition) it to the vacation photos collection.

PPDR4 removed digital object (was removed as part of collection by)

Domain EPDR65 Collection Removal

Range EPDR1 Digital Object

Quantification many to many, necessary

Scope Note This property relates the EPDR65 Collection Removal event with the EPDR1 Digital Object which was removed from a digital collection

Examples

- The dog's photo (EPDR31 Digital Image) *removed as part of collection by* removing (EPDR65 Collection Removal) from the vacation photos collection.

PPDR5 has as repository owner (owns personal digital repository)

Domain EPDR3 Personal Digital Repository

Range E39 Actor

Quantification many to one, necessary

Scope Note This property relates the EPDR3 Personal Digital Repository with E39 Actor who owns it. If the E39 Actor is also an EPDR4 Repository Owner instance then the referenced EPDR3 Personal Digital Repository can be determined to be the current personal digital repository.

Examples

- Mary (E21 Person) *owns personal digital repository* Mary's repository (EPDR3 Personal Digital Repository)
- This personal digital repository (EPDR3 Personal Digital Repository) *has as repository owner* John (EPDR4 Repository Owner)

PPDR6 used as source (was used as source for)

Domain EPDR5 New Version

Range EPDR1 Digital Object

Quantification many to one, necessary

Scope Note This property relates the EPDR5 New Version event with EPDR1 Digital Object from which the new version has been derived.

Examples

- The original business plan (EPDR37 Paginated Document) *was used as source for* a revised version (EPDR5 New Version)

PPDR7 is new version of (is previous version of)

Domain EPDR1 Digital Object

Range EPDR1 Digital Object

Quantification many to many

Scope Note This property relates two EPDR1 Digital Object instances, serving as a shortcut for the more detailed path from EPDR1 Digital Object through PPDR6 used as source (was used as source by) EPDR5 New Version, PPDR81 resulted in (was the result of) to EPDR1 Digital Object. The property also establishes the direction of the relation, with the domain object serving as result and the range object as source (and vice versa in the inverse relation).

Examples

- The color corrected vacation photo (EPDR31 Digital Image) *is new version of* the original one (Digital Image)

PPDR8 acquired digital object (was acquired in)

Domain EPDR7 Ingestion

Range EPDR1 Digital Object

Quantification many to many

Scope Note This property relates the EPDR7 Ingestion event with the EPDR1 Digital Object acquired in said event.

Examples

- The thesis draft (EPDR37 Paginated Document) *was acquired in* the 01/01/2015 backup (EPDR7 Ingestion)

PPDR9 granted current or former ownership to (had current or former ownership by)

Domain EPDR7 Ingestion

Range E39 Actor

Quantification many to one

Scope Note This property relates the EPDR7 Ingestion event with the E39 Actor who took ownership of the acquired digital objects.

Examples

- The ingestion of a downloaded file on 01/01/2015 (EPDR7 Ingestion) *granted current or former ownership to* John (EPDR4 Repository Owner)

PPDR10 has current or former digital owner (is former or current digital owner of)

Domain EPDR1 Digital Object

Range E39 Actor

Superproperty of PPDR11 has current digital owner (is current digital owner of)

Quantification many to many

Scope Note This property relates EPDR1 Digital Object with the E39 Actor that has, or had possession of it. The distinction to PPDR11 has current digital owner (is current digital owner of) is that PPDR10 has current or former digital owner (is former or current digital owner of) does not indicate if the specified owners are current. PPDR10 has current or former digital owner (is former or current digital owner of) is also a shortcut for the more detailed path from EPDR1 Digital Object through PPDR8 acquired digital object (was acquired in), EPDR7 Ingestion PPDR9 granted current or former ownership to (had current or former ownership by) to E39 Actor.

Examples

- The dog photo (EPDR31 Digital Image) *has current or former digital owner* Mary (EPDR21 Person)

PPDR11 has current digital owner (is current digital owner of)

Domain EPDR1 Digital Object

Range EPDR4 Repository Owner

Subproperty of PPDR10 has current or former digital owner (is former or current digital owner of)

Quantification many to one

Scope Note This property relates EPDR1 Digital Object with the EPDR4 Repository Owner that has possession of it, at the time the record is consulted. It serves to formalise the implied relation that the repository owner is also the owner of the content of its personal digital repository. Like its superproperty, it serves as a shortcut to the more detailed path from EPDR1 Digital Object through PPDR8 acquired digital object (was acquired in), EPDR7 Ingestion PPDR9 granted current or former ownership to (had current or former ownership by) to E39 Actor.

Examples

- The dog photo (EPDR31 Digital Image) *has current digital owner* John (EPDR4 Repository Owner)

PPDR12 bequeathed as heirloom (was bequeathed as heirloom in)

Domain EPDR61 Inheritance

Range EPDR1 Digital Object

Subproperty of PPDR8 acquired digital object (was acquired in)

Quantification many to many

Scope Note This property relates the EPDR63 Inheritance event with the EPDR1 Digital Object bequeathed in said event.

Examples

- The family reunion video (EPDR32 Video) *was bequeathed as heirloom in* the 01/01/2015 inheritance (EPDR61 Inheritance)

PPDR13 was inherited by (inherited)

Domain EPDR1 Digital Object

Range E39 Actor

Quantification many to many

Scope Note This property relates the EPDR1 Digital Object with an E39 Actor to which it was bequeathed. It serves as a shortcut to the more detailed path EPDR1 Digital Object through PPDR12 bequeathed as heirloom (was bequeathed as heirloom in) , EPDR61 Inheritance PPDR9 granted current or former ownership to (had current or former ownership by) to E39 Actor

Examples

- The family reunion video (EPDR32 Video) *was inherited by* the Mary (E21 Person)

PPDR14 acquired to repository (content increased by)

Domain EPDR7 Ingestion

Range EPDR3 Personal Digital Repository

Quantification many to many

Scope Note This property relates the EPDR7 Ingestion with an EPDR3 Personal Digital Repository to which digital objects were acquired.

Examples

- John's personal digital repository (EPDR3 Personal Digital Repository) *content increased by* the 01/01/2015 update (EPDR7 Ingestion)

PPDR15 enacted (enacted by)

Domain EPDR10 Policy Enactment

Range EPDR8 Policy

Quantification many to many

Scope Note This property relates the EPDR10 Policy Enactment event with the EPDR8 Policy it enacted.

Examples

- The 01/01/2015 change (EPDR10 Policy Enactment) *enacted* the policy (EPDR8 Policy) that made the dog's photo publicly available.

PPDR16 enacted over (affected by enactment)

Domain EPDR10 Policy Enactment

Range EPDR1 Digital Object

Quantification many to many

Scope Note This property relates the EPDR10 Policy Enactment event with the EPDR1 Digital Object over which the policy has been applied.

Examples

- The 01/01/2015 change (EPDR10 Policy Enactment) has been *enacted over* the dog's photo (EPDR31 Digital Image)

PPDR17 retracted (retracted by)

Domain EPDR11 Policy Retraction

Range EPDR8 Policy

Quantification many to many, necessary

Scope Note This property relates the EPDR11 Policy Retraction event with the EPDR8 Policy that stopped being applied.

Examples

- The 02/01/2015 change (EPDR10 Policy Retraction) *retracted* the policy (EPDR8 Policy) that made the dog's photo publicly available.

PPDR18 retracted over (affected by retraction)

Domain EPDR10 Policy Retraction

Range EPDR1 Digital Object

Quantification many to many, necessary

Scope Note This property relates the EPDR11 Policy Retraction event with the EPDR1 Digital Object over which the policy has stopped being applicable.

Examples

- The 02/01/2015 change (EPDR10 Policy Retraction) has been *retracted over* the dog's photo (EPDR31 Digital Image)

PPDR20 has current policy (currently applies to)

Domain EPDR1 Digital Object

Range EPDR8 Policy

Quantification many to many

Scope Note This property relates the EPDR1 Digital Object with an EPDR8 Policy is currently being applied to it. It also serves as a shortcut to the more detailed path EPDR1 Digital Object through PPDR16 enacted over (affected by enactment), EPDR10 Policy Enactment PPDR15 enacted (enacted by) to EPDR8 Policy.

Examples

- The dog's photo (EPDR31 Digital Image) *has current policy* of not being publicly available (EPDR8 Policy)

PPDR21 has former policy (formerly applied to)

Domain EPDR1 Digital Object

Range EPDR8 Policy

Quantification many to many

Scope Note This property relates the EPDR1 Digital Object with an EPDR8 Policy is has been formally applied to it. It also serves as a shortcut to the more detailed path EPDR1 Digital Object through PPDR18 retracted over (affected by retraction), EPDR11 Policy Retraction PPDR17 retracted (retracted by) to EPDR8 Policy.

Examples

- The dog's photo (EPDR31 Digital Image) *has former policy* of being publicly available (EPDR8 Policy)

PPDR22 communicated (was communicated by)

Domain EPDR23 Communication Attempt

Range EPDR14 Message

Quantification many to one

Scope Note This property relates the EPDR23 Communication Attempt event with the EPDR14 Message instance that was sent in that event.

Examples

- Last night's text (EPDR23 Communication Attempt) *communicated* that Mary was on her way (EPDR18 SMS)

PPDR23 communicated with (received communication in)

Domain EPDR23 Communication Attempt

Range EPDR53 Digital Contact Point

Quantification many to many, necessary

Scope Note This property relates the EPDR23 Communication Attempt event with the EPDR53 Digital Contact Point instances to which the message was sent, thus establishing the recipients of the message.

Examples

- Last night's text (EPDR23 Communication Attempt) *communicated with* the cell phone with number +351 916 900 000 (EPDR55 Phone Number)

PPDR24 communicated from (sent communication in)

Domain EPDR23 Communication Attempt

Range EPDR53 Digital Contact Point

Quantification many to one, necessary

Scope Note This property relates the EPDR23 Communication Attempt event with the EPDR53 Digital Contact Point instance from where a message was sent, thus establishing the message's sender.

Examples

- Last night's text (EPDR23 Communication Attempt) *communicated from* Mary's phone with the number +351 966 900 000 (EPDR55 Phone Number)

PPDR25 sent to (was used to receive)

Domain EPDR14 Message

Range EPDR53 Digital Contact Point

Quantification many to many

Scope Note This property relates the EPDR14 Message with a recipient represented as an EPDR53 Digital Contact Point instance. It serves as a shortcut to the more detailed path EPDR14 Message through PPDR22 communicated (was communicated by), EPDR23 Communication Attempt PPDR23 communicated with (received communication in) to EPDR53 Digital Contact Point.

Examples

- The text message that said that Mary was on her way (EPDR18 SMS) *was sent to* the cell phone with number +351 916 900 000 (EPDR55 Phone Number)

PPDR26 sent from (was used to send)

Domain EPDR14 Message

Range EPDR53 Digital Contact Point

Quantification many to one

Scope Note This property relates the EPDR14 Message with its sender represented as an EPDR53 Digital Contact Point instance. It serves as a shortcut to the more detailed path EPDR14 Message through PPDR22 communicated (was communicated by), EPDR23 Communication Attempt PPDR24 communication from (sent communication in) to EPDR53 Digital Contact Point.

Examples

- Mary's phone with the number +351 966 900 000 (EPDR55 Phone Number) *was used to send* the message that said that she was on her way (EPDR18 SMS)

PPDR27 has next message (has previous message)

Domain EPDR14 Message

Range EPDR14 Message

Quantification one to many

Scope Note This property relates EPDR14 Message instances, establishing a relative order between them without the need to resort to the event path. The quantification in this property allows for branching paths within the messages. This means that the domain (subject) EPDR14 Message can have been been the target of multiple follow-ups (for instance when messages were sent to multiple recipients and each one of them produces a different reply), yet those follow ups can at most refer to a single previous message. Furthermore, it is assumed that each message can only belong to a single conversation.

Examples

- The email with the thesis proposal (EPDR16 Email) *has next message* that came with proposed corrections (EPDR16 Email)

PPDR28 created entry (was created in)

Domain EPDR41 Web Page Visit

Range EPDR40 Browser History Entry

Quantification one to one

Scope Note This property relates the EPDR41 Web Page Visit with the EPDR40 Browser History Entry it created.

Examples

- Visiting google in 01/01/2015 (EPDR41 Web Page Visit) *created entry* in the browser's history (*EPDR40 Browser History Entry*)

PPDR29 visited (was visited in)

Domain EPDR41 Web Page Visit

Range EPDR65 Web Address

Quantification many to one, necessary

Scope Note This property relates the EPDR41 Web Page Visit with the EPDR65 Web Address which was contacted at visit time.

Examples

- The www.google.com page (EPDR65 Web Address) *was visited in* 01/01/2015 *EPDR41 Web Page Visit*

PPDR29 has web page address (is web page address of)

Domain EPDR40 Browser History Entry

Range EPDR65 Web Address

Quantification many to one, necessary

Scope Note This property relates the EPDR40 Browser History Entry with the EPDR65 Web Address which was visited when the entry was created. It serves as a shortcut to the more detailed path EPDR40 Browser History Entry through PPDR28 created entry (was created in), EPDR41 Web Page Visit PPDR29 visited (was visited in) to EPDR65 Web Address.

Examples

- The browser entry (EPDR40 Browser Entry) *has web page address* www.google.com

PPDR30 has acquaintance (is acquaintance in)

Domain EPDR47 Acquaintance Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR47 Acquaintance Relation has or had as a member the specified E39 Actor. Acquaintance relations can be considered to be symmetrical (i.e. all participants acknowledge the existence of the others) and thus there is no need for this property to define a "direction" for the relation.

Examples

- The conference reunion participants (EPDR47 Acquaintance Relation) *has acquaintance* Eve (E21 Person)

PPDR31 is acquainted with

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that a E39 Actor is acquainted with another E39 Actor. It serves as a shortcut to the more detailed path E39 Actor through PPDR30 has acquaintance (is acquaintance in), EPDR47 Acquaintance Relation PPDR30 has acquaintance (is acquaintance in) to E39 Actor. As the implicit EPDR47 Acquaintance Relation can be considered symmetrical, this property does not define a direction for the relation.

Examples

- John (EPDR4 Repository Owner) *is acquainted with* Eve (E21 Person)

PPDR32 has friend in (sees as friends those in)

Domain EPDR48 Friendship Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR48 Friendship Relation has as a member the specified E39 Actor. This member sees the other members of this relation as being its trusted friends, thus defining the direction of the relation as being from the specified E39 Actor towards the other participants.

Examples

- John (EPDR4 Repository Owner) *sees as friends those in* Mary's relation (EPDR48 Friendship Relation)

PPDR33 has been considered friend in (seen as friend by some in)

Domain EPDR48 Friendship Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR48 Friendship Relation has as a member the specified E39 Actor. This member is considered by some of the other members of this relations as being a trusted friend, thus defining the direction of the relation as being from some of the other members towards the specified E39 Actor.

Examples

- Mary *E21 Person* is *seen as friend by some in* Mary's relation (EPDR48 Friendship Relation)

PPDR34 is friend of (seen as friend by)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the domain E39 Actor considers those E39 Actor in the range as trusted friends. Friendship relations are not necessarily symmetrical, which means that to represent a mutual friendship there is the need to employ two pairs of this property, one for each actor, reversing their domain/range position as needed. It serves as a shortcut to the more detailed path E39 Actor through PPPDR32 has friend in (sees as friends those in), EPDR48 Friendship Relation PPDR33 has been considered friend in (seen as friend by some in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is friend of* Mary (E21 Person)

PPDR35 has romantic partner (is involved in)

Domain EPDR49 Romantic Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR48 Romantic Relation has as a member the specified E39 Actor. This member is actively involved with the other participants in the relation at an emotional and or physical level. It should be noted that the EPDR49 Romantic Relation as defined is symmetrical, and thus this property does not define a direction for the relation.

Examples

- John's relation (EPDR44 Romantic Relation) *has romantic partner* John (EPDR4 Repository Owner)

PPDR36 is involved with

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that a E39 Actor is romantically involved with another E39 Actor. It serves as a shortcut to the more detailed path E39 Actor through PPDR35 has romantic partner (is involved in), EPDR49 Romantic Relation PPDR35 has romantic partner (is involved in) to E39 Actor. As the implicit EPDR49 Romantic Relation can be considered symmetrical, this property does not define a direction for the relation.

Examples

- John (EPDR4 Repository Owner) *is involved with* Mary (E21 Person)

PPDR37 has as antagonised (sees as antagonists those in)

Domain EPDR50 Antagonistic Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR50 Antagonistic Relation has as a member the specified E39 Actor. This member dislikes the other members of this relation, though it isn't' necessarily disliked by them. This defines the direction of the relation as being from the specified E39 Actor towards the other participants.

Examples

- Eve's troubled relation (EPDR50 Antagonistic Relation) **has as antagonised** John *EPDR4 Repository Owner*

PPDR38 has as antagonist (seen as antagonist by some in)

Domain EPDR50 Antagonistic Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many, necessary

Scope Note This property establishes that a EPDR50 Antagonistic Relation has as a member the specified E39 Actor. This member is considered by some of the other members of the relation as being hostile to them, though he himself may no be aware of such opinions. The end result is that this defines the direction of the relation as being from some of the other members towards those specified by the use of this property.

Examples

- Eve (E21 Person) is *seen as antagonist by some in* Eve's troubled relation (EPDR50 Antagonistic Relation)

PPDR39 is antagonist of (sees as antagonist)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the domain E39 Actor considers those E39 Actor in the range hostile toward himself. Antagonistic relations are not necessarily symmetrical, which means that to represent mutual hostility there is the need to employ two pairs of this property, one for each actor reversing their domain/range position as needed. It serves as a shortcut to the more detailed path E39 Actor through PPDR37 has been antagonised in (sees as antagonists those in), EPDR50 Antagonistic Relation has been considered antagonist in (seen as antagonist by some in) to E39 Actor.

Examples

- Eve (E21 Person) *is antagonist of* John (EPDR41 Repository Owner)

PPDR40 has as pupil (sees as mentors those in)

Domain EPDR51 Mentor Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR51 Mentor Relation has as a member the specified E39 Actor. This member sees the other participants of this relation as tutor figures, though they do not necessarily see him as a student. This defines the direction of the relation as being from the specified E39 Actor towards the other participants. The presence of this property also precludes the presence of the PPDR42 is mentor in (sees as pupils those in) property.

Examples

- Paul's educational relation *has as pupil* John (EPDR4 Repository owner)

PPDR41 considers as mentor (seen as mentor by some in)

Domain EPDR51 Mentor Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR51 Mentor Relation has as a member the specified E39 Actor. This member is considered by some of the other members of the relation as a tutor, though he himself may not be aware of those opinions. The end result is that this defines the direction of the relation as being from some of the members towards those specified by the use of this property.

Examples

- Paul's educational relation (EPDR51 Mentor Relation) *considers as mentor* Paul (E21 Person)

PPDR42 has as mentor (sees as pupils those in)

Domain EPDR51 Mentor Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR51 Mentor Relation has as a member the specified E39 Actor. This member sees himself as a tutor figure to the other members of the relation, though they do not necessarily see him as such. This defines the direction of the relation as being from the specified E39 Actor towards the other participants. The presence of this property also precludes the presence of the PPDR40 is pupil in (sees as mentors those in).

Examples

- John's educational relation (EPDR51 Mentor Relation) *has as mentor* John (EPDR4 Repository Owner)

PPDR43 considers as pupil (seen as pupil by some in)

Domain EPDR51 Mentor Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note this property establishes that EPDR51 Mentor Relation has as a member the specified E39 Actor. This member is considered by some of the other members of the relation as a pupil, though he himself may not be aware of such. The end result is that this defines the direction of the relation as being from some of the members towards those specified by the use of this property.

Examples

- John's educational relation (EPDR51 Mentor Relation) *considers as pupil* Peter (E21 Person)

PPDR44 is pupil of (seen as mentor by)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the domain E39 Actor considers himself as a pupil to those E39 Actor defined in the range. Mentor relationships are not necessarily symmetrical, which means that to represent a mutual relation there is the need to also employ the PPDR45 is mentor of (seen as pupil by) to represent the relation in the other direction. It serves as a shortcut to the more detailed path E39 Actor through PPDR40 has as pupil (sees as mentors those in), EPDR51 Mentor Relation PPDR41 considers as mentor (seen as mentor by some in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is pupil of* Paul (E21 Person)

PPDR45 is mentor of (seen as pupil by)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the domain E39 Actor considers himself as a mentor to those E39 Actor defined in the range. Mentor relationships are not necessarily symmetrical, which means that to represent a mutual relation there is the need to also employ the PPDR44 is pupil of (seen as mentor by) to represent the relation in the other direction. It serves as a shortcut to the more detailed path E39 Actor through PPDR42 has as mentor (sees as pupils those in), EPDR51 Mentor Relation PPDR43 considers as pupil (seen as pupil by some in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is mentor of* Peter (E21 Person)

PPDR46 has coworker in (is coworker in)

Domain EPDR52 Work Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR51 Work Relation has as a member the specified E39 Actor. This member cooperates with the others members of a relation, regardless of hierarchical position. It should be noted that EPDR51 Work Relation is by nature a symmetrical relation and thus this property does not define the direction of the relation.

Examples

- MetaBusiness relation *has coworker in* John (EPDR4 Repository Owner)

PPDR46 is coworker of

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in a work relation (which are symmetrical by nature). This property serves as a shortcut to the more detailed path E39 Actor through PPDR46 has coworker in (is coworker in), EPDR52 Work Relation PPDR46 has coworker in (is coworker in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is coworker of* Paul

PPDR47 has parent in (is parent in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as parent figure (mother, stepmother, father or stepfather). This property establishes the direction of the relation as being from the other member of the relation to the one specified by this property.

Examples

- John's father family relation (EPDR46 Family Relation) *has parent in* Noah (E21 Person)

PPDR48 has mother in (is mother in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR47 has parent in (is parent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to server as parent figure, specifically a biological mother, thus establishing the direction of the relation as being from the other member of the relation to the one specified by this property. The primary difference between this and the PPDR47 has parent in (is parent in) property is the implicit gender definition that comes with this property.

Examples

- John's mother family relation (EPDR46 Family Relation) *has mother in* Emma (E21 Person)

PPDR49 has stepmother in (is stepmother in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR47 has parent in (is parent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to server as parent figure, specifically a mother figure such as a stepmother, thus establishing the direction of the relation as being from the other member of the relation to the one specified by this property. The primary difference between this and the PPDR47 has parent in (is parent in) property is the implicit gender definition that comes with this property.

Examples

- Mary's mother family relation (EPDR46 Family Relation) *has stepmother in* Olivia (E21 Person)

PPDR50 has father in (is father in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR47 has parent in (is parent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to server as parent figure, specifically a biological father, thus establishing the direction of the relation as being from the other member of the relation to the one specified by this property. The primary difference between this and the PPDR47 has parent in (is parent in) property is the implicit gender definition that comes with this property.

Examples

- John's father family relation (EPDR46 Family Relation) *has father in* Noah (E21 Person)

PPDR51 has stepfather in (is stepfather in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR47 has parent in (is parent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to server as parent figure, specifically a stepfather, thus establishing the direction of the relation as being from the other member of the relation to the one specified by this property. The primary difference between this and the PPDR47 has parent in (is parent in) property is the implicit gender definition that comes with this property.

Examples

- Mary's father family relation (EPDR46 Family Relation) *has stepfather in* Ethan (E21 Person)

PPDR52 has child in (is child in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a child (which here refers to being an offspring, regardless of being blood related or not, and not the the age of the referred E39 Actor). This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation.

Examples

- John's father family relation (EPDR46 Family Relation) *has child in* John (EPDR4 Repository Owner)

PPDR53 has son in (is son in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has child in (is child in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a son. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation. The primary difference between this and the PPDR52 has child in (is child in) property is the implicit gender definition that comes with this property.

Examples

- John's father family relation (EPDR46 Family Relation) *has son in* John (EPDR4 Repository Owner)

PPDR54 has daughter in (is daughter in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has child in (is child in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a daughter. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation. The primary difference between this and the PPDR52 has child in (is child in) property is the implicit gender definition that comes with this property.

Examples

- Mary's father family relation (EPDR46 Family Relation) *has daughter in* Mary (E21 Person)

PPDR55 is child of (is parent of)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in a family relationship, specifically that one of the participants is a child of the other participant (where child refers to being an offspring, not to the participant's age). It also serves as a shortcut to the more detailed path E39 Actor through PPDR52 has child in (is child in), EPDR46 Family Relation PPDR47 has parent in (is parent in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is child of* Noah (E21 Person)

PPDR56 has sibling in (is sibling in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a sibling to the other member of the relation. Since being a sibling is effectively a symmetrical relation, this property does no define a direction for the relation.

Examples

- John's sibling relation (EPDR46 Family Relation) *has sibling in* Jacob (E21 Person)

PPDR57 has brother in (is brother in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has sibling in (is sibling in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a biological brother to the other member of the relation. Since being a brother is effectively a symmetrical relation, this property does no define a direction for the relation, with the main difference for the PPDR52 has sibling in (is sibling in) property being the implicit gender connotation.

Examples

- John's sibling relation (EPDR46 Family Relation) *has brother in* Jacob (E21 Person)

PPDR58 has sister in (is sister in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has sibling in (is sibling in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a biological sister to the other member of the relation. Since being a brother is effectively a symmetrical relation, this property does no define a direction for the relation, with the main difference for the PPDR52 has sibling in (is sibling in) property being the implicit gender connotation.

Examples

- John's sibling relation (EPDR46 Family Relation) *has sister in* Mia (E21 Person)

PPDR59 has stepbrother in (is stepbrother in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has sibling in (is sibling in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a stepbrother to the other member of the relation. Since being a stepbrother is effectively a symmetrical relation, this property does no define a direction for the relation, with the main difference for the PPDR52 has sibling in (is sibling in) property being the implicit gender connotation.

Examples

- Mary's sibling relation (EPDR46 Family Relation) *has stepbrother in* Michael (E21 Person)

PPDR60 has stepsister in (is stepsister in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR52 has sibling in (is sibling in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a stepsister to the other member of the relation. Since being a stepsister is effectively a symmetrical relation, this property does no define a direction for the relation, with the main difference for the PPDR52 has sibling in (is sibling in) property being the implicit gender connotation.

Examples

- Mary's sibling relation (EPDR46 Family Relation) *has stepsister in* Emily (E21 Person)

PPDR62 is sibling of

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in an implicit family relationship, specifically that the linked participants are siblings. Since being a sibling is effectively a symmetrical relation, this property does not define a direction for the relation, though it serves as a shortcut to the more detailed path E39 Actor through PPDR56 has sibling in (is sibling in), EPDR46 Family Relation PPDR56 has sibling in (is sibling in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is sibling of* Mia (E21 Person)

PPDR61 has uncle/aunt in (is uncle/aunt in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a uncle or aunt to the other member of the relation. It should be noted that there is no standard gender neutral designation for this family relation, and thus a composite of both gender's designation is used in here. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property.

Examples

- John's relatives relation (EPDR46 Family Relation) *has uncle/aunt in* Charlotte (E21 Person)

PPDR62 has uncle in (is uncle in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR61 has uncle/aunt in (is uncle/aunt in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a uncle to the other member of the relation. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property. The primary difference between this and the PPDR61 has uncle/aunt in (is uncle/aunt in) property is the implicit gender definition that comes with this property.

Examples

- John's relatives relation (EPDR46 Family Relation) *has uncle in* Alexander (E21 Person)

PPDR63 has aunt in (is aunt in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR61 has uncle/aunt in (is uncle/aunt in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a uncle/aunt to the other member of the relation. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property. The primary difference between this and the PPDR61 has uncle/aunt in (is uncle/aunt in) property is the implicit gender definition that comes with this property.

Examples

- John's relatives relation (EPDR46 Family Relation) *has aunt in* Alexander (E21 Person)

PPDR65 has nephew/niece in (is nephew/niece in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a nephew or nice to the other member of the relation. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation. The primary difference between this and the PPDR61 has uncle/aunt in (is uncle/aunt in) property is the implicit gender definition that comes with this property.

Examples

- John's relatives relation (EPDR46 Family Relation) *has nephew/niece in* John (EPDR4 Repository Owner)

PPDR66 has nephew in (is nephew in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR65 has nephew/niece in (is nephew/niece in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a nephew to the other member of the relation. It should be noted that there is no standard gender neutral designation for this family relation, and thus a composite of both gender's designation is used in here. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation.

Examples

- John's relatives relation (EPDR46 Family Relation) *has nephew in* John (EPDR4 Repository Owner)

PPDR67 has niece in (is niece in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR65 has nephew/niece in (is nephew/niece in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a nephew or nice to the other member of the relation. It should be noted that there is no standard gender neutral designation for this family relation, and thus a composite of both gender's designation is used in here. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation.

Examples

- Mary's relatives relation (EPDR46 Family Relation) *has niece in* Mary (E21 Person)

PPDR67 is nephew/niece of (is uncle/aunt of)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in an implicit family relation, specifically that the linked participants are respectively nephew/niece and uncle/aunt of each other. It serves as a shortcut to the more detailed path E39 Actor through PPDR65 has nephew/niece in (is nephew/niece in), EPDR46 Family Relation PPDR61 has uncle/aunt in (is uncle/aunt in) to E39 Actor.

Examples

- John (EPDR4 Repository owner) *is nephew/niece of* Alexander (E21 Person)

PPDR68 has cousin in (is cousin in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a cousin to the other member of the relation. Since being a cousin is effectively a symmetrical relation, this property does not define a direction for the relation.

Examples

- John cousin relation (EPDR46 Family Relation) *has cousin in* Rebecca (E21 Person)

PPDR69 is cousin of

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in an implicit family relation, specifically that the linked participants are cousins. Since being a cousin is effectively a symmetrical relation, this property does not define a direction for the relation, though it serves as a shortcut to the more detailed path E39 Actor through PPDR68 has cousin in (is cousin in), EPDR46 Family Relation PPDR68 has cousin in (is cousin in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is cousin of* Rebecca (E21 Person)

PPDR70 has grandparent in (is grandparent in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a grandparent to the other member of the relation. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property.

Examples

- John's grandparent relation (EPDR46 Familiy Relation) *has grandparent in* Ava (E21 Person)

PPDR71 has grandfather in (is grandfather in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR70 has grandparent in (is grandparent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a grandfather to the other member of the relation. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property. The primary difference between this and the PPDR70 has grandparent in (is grandparent in) property is the implicit gender definition that comes with this property.

Examples

- John's grandparent relation (EPDR46 Familiy Relation) *has grandparent in* Luke (E21 Person)

PPDR72 has grandmother in (is grandmother in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR70 has grandparent in (is grandparent in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a grandmother to the other member of the relation. This property establishes the direction of the relation as being from the other member of the relation to the member specified by this property. The primary difference between this and the PPDR70 has grandparent in (is grandparent in) property is the implicit gender definition that comes with this property.

Examples

- John's grandparent relation (EPDR46 Familiy Relation) *has grandmother in* Ava (E21 Person)

PPDR73 has grandchild in (is grandchild in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of P107 has current or former member (is current or former member of)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a grandchild to the other member of the relation. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation.

Examples

- John's grandparent relation (EPDR46 Familiy Relation) *has grandchild in* John (EPDR4 Repository Owner)

PPDR74 has grandson in (is grandson in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR73 has grandchild in (is grandchild in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a grandson to the other member of the relation. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation. The primary difference between this and the PPDR73 has grandchild in (is grandchild in) property is the implicit gender definition that comes with this property.

Examples

- John's grandparent relation (EPDR46 Familiy Relation) *has grandson in* John (EPDR4 Repository Owner)

PPDR75 has granddaughter in (is granddaughter in)

Domain EPDR46 Family Relation

Range E39 Actor

Subproperty of PPDR73 has grandchild in (is grandchild in)

Quantification many to many

Scope Note This property establishes that a EPDR46 Family Relation has as a member the specified E39 Actor. The role of this member in the relation is to serve as a granddaughter to the other member of the relation. This property establishes the direction of the relation as being from the member specified by this property to the other member of the relation. The primary difference between this and the PPDR73 has grandchild in (is grandchild in) property is the implicit gender definition that comes with this property.

Examples

- Mary's grandparent relation (EPDR46 Familiy Relation) *has granddaughter in* Mary (E21 Person)

PPDR76 is grandchild of (is grandparent of)

Domain E39 Actor

Range E39 Actor

Quantification many to many

Scope Note This property establishes that the E39 Actor instances it links are engaged in an implicit family relation, specifically that the linked participants are grandchild and grandparent to each other. This property serves as a shortcurt to the more detailed path E39 Actor through PPDR73 has grandchild in (is grandchild in), EPDR46 Family Relation PPDR70 has grandparent in (is grandparent in) to E39 Actor.

Examples

- John (EPDR4 Repository Owner) *is grandchild of* Ava (E21 Person)

PPDR77 deprecated contact (was deprecated in)

Domain EPDR58 Contact Deprecation

Range EPDR53 Digital Contact Point

Quantification many to many

Scope Note This property relates the EPDR58 Contact Deprecation event with the EPDR53 Digital Contact Point instance that was deprecated in that event.

Examples

- The phone number +351 996 000 002 (EPDR55 Phone Number) *was deprecated in* 01/01/2015 (EPDR58 Contact Deprecation)

PPDR78 has replacement contact (became replacement contact in)

Domain EPDR59 Contact Replacement

Range EPDR53 Digital Contact Point

Quantification many to many

Scope Note This property relates the EPDR59 Contact Replacement event with the EPDR53 Digital Contact Point instance that has replaced the previous EPDR53 Digital Contact Point.

Examples

- The phone number +351 996 000 003 (EPDR55 Phone Number) *became replacement contact in* 01/01/2015 (EPDR58 Contact Deprecation)

PPDR79 has current contact point (currently provides access to)

Domain E39 Actor

Range E51 Contact Point

Subproperty of P76 has contact point (provides access to)

Quantification many to many

Scope Note This property identifies an E51 Contact Point that can be used to contact an E39 Actor. The difference between this and its superproperty is that this property establishes that the identified E51 Contact Point is one that can currently be used to contact the E39 Actor, as opposed to one that is deprecated (i.e a former contact point).

Examples

- Mary (E21 Person) *has current contact point* the phone with the number +351 966 900 000 (EPDR55 Phone Number)

PPDR80 has current or former credentials (is current or former credential for)

Domain EPDR60 Digital Account

Range EPDR61 Digital Account Credentials

Quantification many to many

Scope Note This property identifies an EPDR61 Digital Account Credentials that can be used to access and take control of an EPDR60 Digital Account.

Examples

- John's email account (EPDR60 Digital Account) *has current or former credentials* john@example.com//12345 (EPDR62 UsernamePassword Credentials)

PPDR81 resulted in (was the result of)

Domain EPDR5 New Version

Range EPDR1 Digital Object

Quantification many to one, necessary

Scope Note This property relates the EPDR5 New Version event with EPDR1 Digital Object that was created as result of said event.

Examples

- The revised business plan (EPDR37 Paginated Document) *was the result of* a revised version (EPDR5 New Version)

4.1.3 Examples

The Figure 4.3 shows an example of how the previously detailed extensions to the CIDOC/CRM model can be used to reason about event information when dealing with digital objects. In it, entities are represented as rectangles (light blue ones for native CIDOC/CRM entities, light green ones for those who belong to the proposed extension). Double arrows are used to indicate *isA* relationships between entities while single arrows represent the properties that connect them, being that the labels for "shortcut" properties appear with a grey background.

The diagram itself represents a single ingestion event of an SMS Message instance. Though a single SMS message carries with it a limited amount of information (typically 160, 140 or 70 characters depending on the character encoding used), their surrounding context can bring

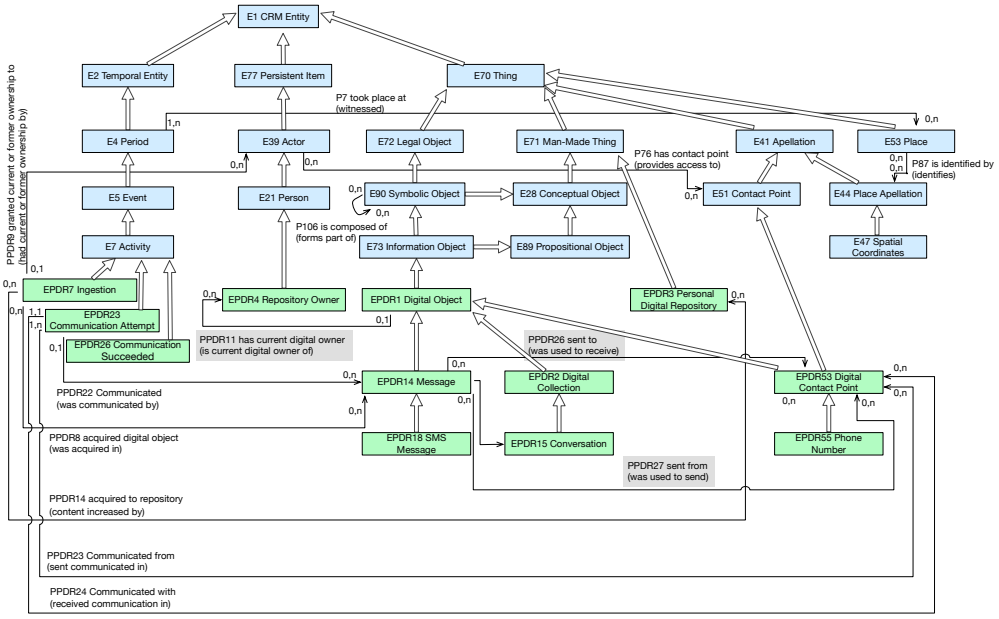


Figure 4.3: Reasoning about the ingestion event of an SMS Message instance

with it a significant amount of additional spatial, temporal and social information. Spatial information can convey the location of the device (and with it of the repository owner) at the time the time the message was received. This information can be expressed through the use of the CIDOC/CRM property P7 took place at (witnessed) to connect a period instance (of which the EPDR23 Communication Attempt is a particular case through inheritance) with an instance of the E53 Place entity, which in turn is used to identify a physical place, for instance by resorting to its coordinates represented through an instance of the E47 Spatial Coordinates entity. It should be noted that coordinates can often (though not always) be resolved to other, more human friendly, forms of place appellation (such as instances of the E45 Address entity, which is not represented in the diagram), with the use of reverse geocoding services, yet such resolution requires that either the personal digital repository or the gathering agent has access to one of those services to perform it. Temporal information about the EPDR12 SMS Message instance is mainly carried through the events in which it is involved, with the use of instances of the E52 Time-Span entity, linked by P4 has time-span (is time-span of) property (both of which are not represented in the diagram). In addition to temporal information, events act as an hub for other entities. For instance, from the EPDR23 Communication Attempt event it is possible to discover a message's sender and receiver (represented by instances of the E39 Actor entity) by way of the E51 Contact Point instances, or as in this case through EPDR55 Phone Number, one of the E51 Contact Point derived entities. The same thing can be achieved from the EPDR12 SMS Message instance itself by way of the PPDR26 sent to (was used to receive) "shortcut" property, with the caveat that once again it connects the message with the contact point who send it, not directly with the actor that can be reached by using that contact point. The EPDR18 SMS Message instances that are part of an EPDR15 Conversation instance can be conveyed through the use of the CIDOC/CRM property P106 is composed of (forms part of). This happens since instances of both of those entities can also

be considered instances of E90 Symbolic Object through inheritance. It should be noted that a possible alternative to describe this relation would be through the use of the CIDOC/CRM property P148 has component (is component of), though on that case it would be applied as instances of both entities are also instances of E89 Propositional Object through inheritance. As a final remark it should also be noted that the ontology does not specify all of the possible EPDR1 Digital Objects' derivations nor all of the possible properties that can be applied to them. Instead, the definition of most classes and properties is only precise enough to describe its form and function in the personal digital repository, while allowing (and even encouraging) further specialisation, that can come from either additional extension provided by repository modules or from existing specialised vocabularies (for instance the EXIF ontology for image metadata), with the corresponding mappings also being provided by the repository modules that use them. These mappings can take advantage of the type system already in place in the CIDOC/CRM model, in which nearly every entity defined in the model (with the exception of E59 Primitive Value) can use the property P2 has type (is type of) to further clarify what is the type of the entity, and which can be used as a bridge to the more specific external ontologies or vocabularies.

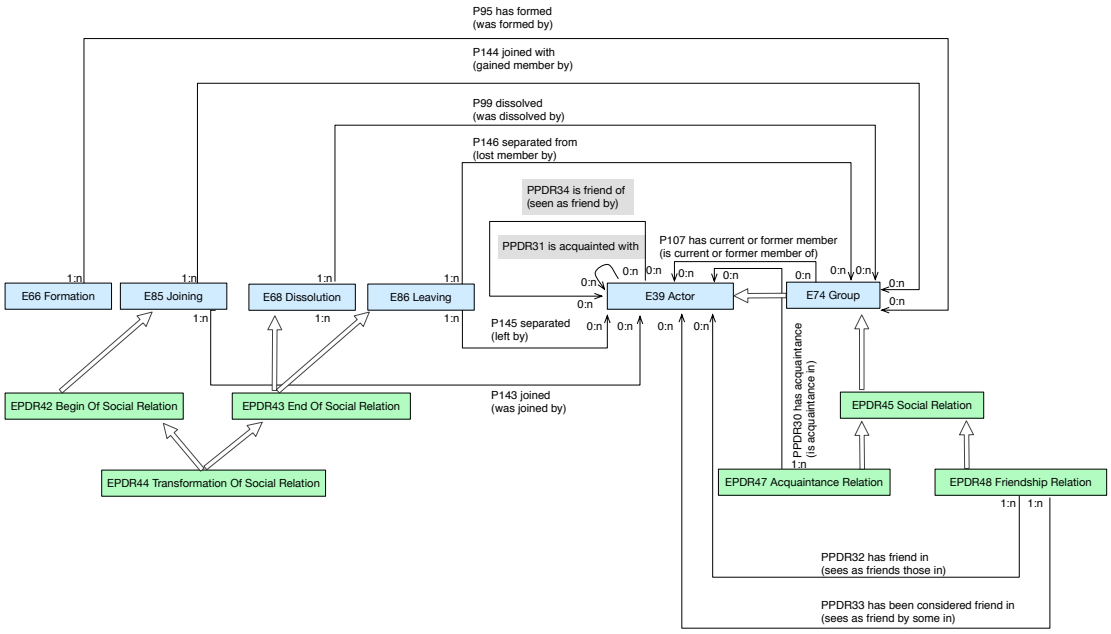


Figure 4.4: Reasoning about social relations

The diagram in Figure 4.4 shows an example of how the previously detailed extensions to the CIDOC/CRM model can be used to reason about social relations. Unlike the previous diagram, this one has been simplified by omitting most part of the entity hierarchy already defined by the core CIDOC/CRM model. The diagram itself focus on two types of social relations, acquaintance and friendship relations. Following the model's hint, relationships are modelled as being derived from the E74 Group entity. Like members in a group, the participants in a social relation can have different roles, which are encoded in the properties used to link instances of participant E39 Actor with the instances of the relations themselves, with the proposed extended properties being themselves derived from the already existing

CIDOC/CRM property P107 has current or former member (is current or former member of), thus retaining the ability to convey that group membership. In an EPDR47 Acquaintance Relation instance, the participants have the same role in the relation, regardless of the perspective from which they are seen. This is represented by only having a single property, PPDR30 has acquaintance (is acquaintance in) that is used to relate the participants to that relation type. On the other hand, the participants in an EPDR48 Friendship Relation instance have distinct roles. Participants can either see the other participants as friends, or be considered as a friend by other participants. These roles are expressed by the use of the PPDR32 has friend in (sees as friends those in) or the PPDR33 has been considered friend in (sees as friend by some in) property to indicate their membership within a relation. Instances of the EPDR48 Friendship Relation entity can be bypassed with the use of of a "shortcut" property, PPDR34 is friend of (seen as friend by), which encodes both possible roles within the inverse relation. Thus, using the PPDR34 is friend of (seen as friend by) to state that an E39 Actor instance is a friend of second, distinct E39 Actor instance also states that the second instance is seen as a friend by the first one. Such behaviour implies that some social relations are directional, and thus would need an additional relation made with the PPDR34 is friend of (seen as friend by) property, but with the original subject and object reversed in order to express a true mutual friendship. As with other parts of the proposed extension, temporal information regarding social relations can be conveyed through the use of events. Defined events are focused on the three major events that are probable to affect a social relation, which means that they represent its initial point with the use of instances of the EPDR42 Begin Of Social Relation entity, its final point with the use of instances of the EPDR43 End Of Social Relation and notable changes that are capable of transforming one relation type into another with the use of instances of the EPDR44 Transformation Of Social Relation entity. These entities are meant to represent a combination of already existing events that affect groups. Thus an EPDR42 Begin Of Social Relation entity is derived both from the E66 Formation and E85 Joining entities, while an EPDR43 End Of Social Relation entity is derived from E68 Dissolution and E85 Leaving entities. This is done since the seminal event that creates a social relation also binds its participants to join it, and conversely the event that ends a social relation also implies that the participants leave said relation. Furthermore this strategy also allows the application of the existing CIDOC/CRM properties to specify whom has joined and which relation have their joined (for example, the P95 has formed (was formed by) and P144 joined with (gained member by) properties). It should be noted that although the proposed extensions have the capability to represent several types of relations and the events that lead to their creation, transformation or demise, this does not mean that it will be immediately possible for content collection tools to actually infer the existence of those social relations or of the events that accompany them. This is one of the domains where it is most notorious that the information and content gathered is probably going to be incomplete, be it because the events already came to pass, because they didn't left any digital evidence or any other reason.

As previously mentioned, the proposed ontology does not cover all possible cases, nor it defines all possible entity derivations or properties. It instead relies on other extensions to deal with specialised content types and their associated metadata, as well as mappings. Mappings provide the instructions on how to translate existing metadata schemas elements so that they can be represented by the entities and properties provided by CIDOC/CRM, and by extension the proposed ontology for personal digital repositories. One of the schemas that is already mapped onto the CIDOC/CRM is Dublin Core [32, 33]. Using those previously existing mappings allows the proposed ontology to avoid having to define extra properties and entities

to represent elements from DC. On the other hand, elements that come from other medata sources (for instance metadata from specific file formats, such as PDF or Word Documents) can often be interpreted in terms of what is defined by DC. This means that effectively it is also possible to map and represent some elements from those sources without having to define extra properties and entities. An example of how to map a dublin cored encoded record about a technical report can be found in the CIDOC/CRM tutorial [31], with the end results being illustrated in Figure 4.5. If this example would be applied to content submitted to the personal digital repository, the major change would come from the E33 Linguistic Object entity, that would be changed to the appropriate derivation of an EPDR1 Digital Object, if enough information is provided to determine which one to use, or the base EPDR1 Digital Object itself otherwise. Such change has no impact on most of applied properties, as they can be applied to either E1 CRM Entity, E28 Conceptual Object or E70 Thing, which are all part of the inheritance tree for EPDR1 Digital Object. The one exception to this is the P72 has language (is language of), which has in its scope set to E33 Linguistic Object. Not all EPDR1 Digital Object instances are bound to carry with them linguistic information expressed in a natural language, and even those derivations intended to represent documents in whose information is expressed in natural language provide no warranty that they do so in every case. As such that particular property cannot be directly applied to the EPDR1 Digital Object. Nevertheless, should the need arise there is nothing preventing the extension of the provided entities with those that also include E33 Linguistic Object in their inheritance chain in order to be able to use that property directly. Until then a possible alternative is the use of either an P3 has note to convey the additional information, or transform it into am instance of a E55 Type and represent it through the type system.

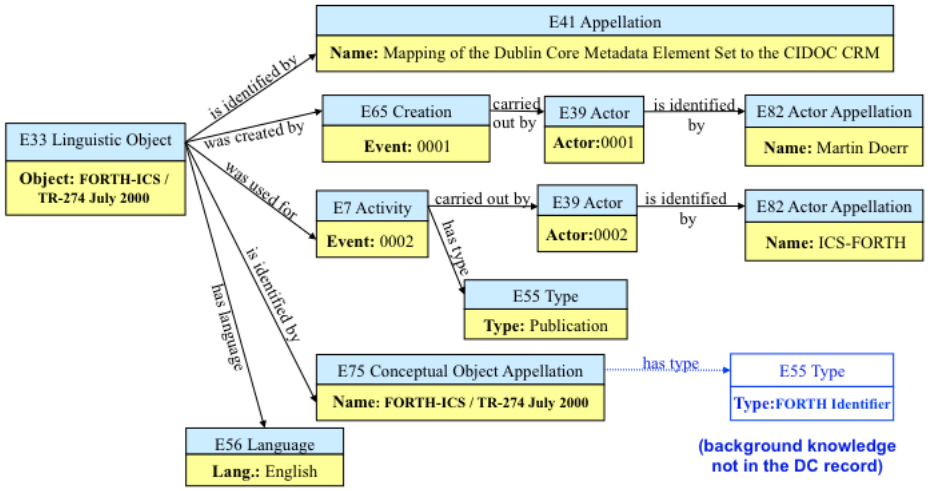


Figure 4.5: Example of DC/CIDOC mapping [31]

4.2 Chapter Conclusions

This chapter describes a possible ontology that can be used to support the personal digital repository's shared context. Personal scenarios are bound to yield different types of content and digital objects to be collected in the personal digital repository. Each type of content brings with it its own set of metadata, being that some of the metadata from different content types might be functionally the same but with a different representation or organisation scheme. For the personal digital repository's shared context to be able to do its job (i.e. to serve as canvas where different pieces of content can become interconnected) there is the need to normalise some of that received metadata into a workable representation to reduce the risk of introducing into the shared context ambiguous (or even worst strait up incompatible) organisations schemes. Given the personal nature of the repository the most obvious approach to create a solution for this issue would be to recruit the repository owner itself to design the organisation scheme that would underpin the shared context. Yet such an approach comes with some downsides such as not every potential user of a personal digital repository will have the expertise to create their own organisation scheme, even in the context of a guided process. Additionally it can be difficult for individuals to assess their interests, be them their past or current interests, let alone to attempt to predict their future interests, and these issues will have a direct impact on how the organisation scheme is created and how it will evolve. Finally, while the adoption of such a strategy would make for truly personal repositories, it would also mean that they would be largely incompatible between them. A less personal, but arguably more workable approach, that was the one selected for the personal digital repository, is to establish a baseline organisation scheme, that can be extended as needed and that must be able to encode within itself additional organisation overlays, being that those organisation overlays are the ones actually defined by the repository owner.

Instead of designing an entire organisation scheme from the ground up, the adopted strategy for the personal digital repository was to adapt an already existing model, with the chosen one being the CIDOC/CRM model. This model has several traits that make it a suitable candidate for adaptation into personal scenarios, such as having a strong emphasis on temporal information for context building, being purposely designed to be extended and being designed to encode within itself user defined hierarchies. However, as a model originally designed for information interchange between memory institutions regarding their collections (that are comprised primarily of physical objects), it lacked concepts relevant to describe digital objects and personal scenarios. Taking advantage of its extensibility, the ontology presented in this chapter is framed as being one possible extension of the CIDOC/CRM model. Its proposed entities and properties are thought to support the digital and personal aspects that are missing from the core mode, both from the interaction point of view (i.e. with additional events that tie into the temporal aspect of the model) as well as with base entities needed to represent digital content. The proposed extension was designed to be itself extensible, as due to the unpredictable nature of personal scenarios there is no way to encompass in a single vocabulary all the entities and properties needed to represent all content types. In the personal digital repository that task is left for the repository's content modules, that in addition to use the base extension presented in this chapter can introduce their own extensions, with the caveat that they should be rooted in the CIDOC/CRM and the proposed extension. Another design trait of the proposed extension is that when possible it eschews the use of the CIDOC/CRM type system. This resulted in having several concepts that could otherwise be modelled as an sub entity (or even as controlled vocabulary term) of the CIDOC/CRM E55 Type entity being

explicitly defined as an entity in the proposed ontology. The CIDOC/CRM model considers both approaches as legitimate, with the proposed rule of thumb for deciding between them being if the underlying concept is stable enough to warrant its own entity. In this case, the motivation to model the concepts as entities came from the need to leave the type system free for use in user defined classification schemes that can be overlaid on top of the one proposed in the ontology defined in this chapter. Of the various proposed extensions with which it attempts to enhance the CIDOC/CRM model, the ones that deal with personal relationships are particularly noteworthy because it is an area where the base CIDOC/CRM is arguably underspecified, with this underspecification having a direct impact in the description of ethnographic collections that need to represent relations between persons other than the biological parent-child one. The standard approach within CIDOC/CRM to model relations is to join the participants in a group. While flexible, this approach also entails (from the group definition) that those that participate in a social relation will be able to act in unison, which is not always the case for all possible relationship types. This in turn has lead to the discussion of alternative approaches that may even pass through the introduction of specific entities [120] and events to describe top level arbitrary relations without such constrain. As an interim solution the proposed extension is still based on the group membership concept, though it creates sub entities of the E74 Group to constrain and further specify the relation type and meaning, while it also introduces events that can serve as "triggers" for the establishment of a relation. Additionally it incorporates the concept of direction in a relationship, in order to represent relations that are not reciprocal (i.e. those in which not all participants may not be aware that they are in a relation), that can be established by the properties used to link each participant to the relation itself. The end result is that the social relation model proposed can be used to represent evolving relations, that morph from one type to another through time with the caveat that the representation is biased by the observer's point of view (which in this case means by the repository owner). It should also be noted that, like all of the defined entities and events, its use will depend on the available data. For instance, without direct intervention from the repository owner it might be difficult or downright impossible to know when an frequently contacted acquaintance has become a friend.

Chapter 5

Personal Digital Repository Reference Implementation

"Nothing will be forever gone
Memories will stay and find their way
What goes around will come around
Don't deny your fears
So let them go and fade into light
Give up the fight here"

Epica

Chasing the Dragon in The Divine Conspiracy, 2007

The previous chapters were dedicated to the description of the architecture and underlying ontology that can be used to build a personal digital repository. The following sections of this chapter will describe a reference implementation of those components and of the content collection tools that can be used to provide content for the reference implementation of a personal digital repository.

5.1 Personal Digital Repository Core

As stated in the Architecture chapter, a personal digital repository forms an intertwined ecosystem that relies on content collection tools to acquire content. On the other hand digital preservation by itself is a low priority issue for most individuals, which means that attempting to get them to use those content collection tools will require that they provide additional benefits in addition to their preservation goals. A possible approach to increase the adoption and use of content collection tools by individuals is to have them provide other services that might be of more immediate use for individuals. Some of those services, particularly those who will act as information integrators will require content or metadata that is not readily available locally, but that may exist as part of the shared context of a personal digital repository. This provides an additional incentive for personal digital repositories to behave in such a way that they remain permanently connected and accessible to both content collection tools and

their owner. In the proposed architecture this incentive is catered to by specifying that a personal digital repository is to be developed as a web application (which by definition already incorporates the need to be accessible in order to be useful). This approach comes with the benefit of shifting direct interaction (i.e. those that are not mediated by content collection tools) between the owner and his personal digital repository from a native application to the browser, an environment that should be (hopefully) familiar to most users in personal scenarios. However this approach comes with the downside that it requires the use of an application server which must be installed and configured by the repository owner. To mitigate this issue, the personal digital repository can be distributed bundled with an embedded version of an application server or be offered as a hosted service to its owners.

As the architecture mandates the personal digital repository to be a web application, the reference version of the personal digital repository was developed with the use of the Grails web application framework [121]. This framework was chosen for being open source, cross platform (it is based on Groovy [122], a language for the Java [123] Virtual Machine, which means that it can be deployed in several application servers that are available for multiple major operative systems) and extensible. Applications developed in this framework can take advantage of the framework's extensibility allowing them to be developed by incorporating existing components which are packaged as plugins. The plugins themselves are self contained slightly modified versions of the applications created with the framework and throughout version 1 and 2 of the Grails application framework are distributed in source format to be added to other web applications at compile time. Despite this, since plugins are essentially self contained web applications they can still be compiled by themselves. The reference implementation of the personal digital repository can use this mode to fulfil the requirement for a modular repository specified by the previously proposed architecture for a personal digital repository. In the reference implementation, Grails' plugins compiled as web applications can be uploaded to the running repository, which stores them. Plugins do not become immediately active, but instead need to be explicitly activated by the repository owner. This is a design consideration that comes from a limitation imposed by the framework itself. While it would be possible to add a newly updated plugin to the repository's reference implementation' classpath (as the framework runs in the Java Virtual Machine), due to the internal workings of the framework artefacts (i.e. web pages, controllers, services, etc.) are only recognised and loaded when a web application is initially loaded. This leads to the need to restart the web application itself to recognise newly added plugins. In the reference implementation this is accomplished by directly calling the underlying application server (Apache Tomcat [124]). For now this strategy effectively restricts the application server that can be used to host the reference implementation of the personal digital repository. Another issue that arises from reusing the framework's plugin system to build the module system for this reference implementation of the personal digital repository is that once installed plugins become globally available. While not a problem for self-hosted deployments that serve a single individual, it may be problematic when attempting to offer the personal digital repository as a service, since plugins installed and used in one repository would automatically become part of all others in the shared environment, even if they are of no use to the other repository owners. If one take into account that there is a mandatory application reload when installing a new plugin this could lead to unpredictable service disruptions. Thus if the reference implementation of the personal digital repository is to be used in a SaaS scenario, it is preferable to isolate it and provide virtual machines to each user instead of attempting to serve multiple users from the same deployed version.

As stated in the Architecture chapter, a personal digital repository uses two types of

metadata storage: one for its digital object catalogue and assorted administrative data and a second one for the digital object's shared context. To store its digital object catalogue and administrative data the reference implementation of the personal digital repository uses MongoDB [125] instead of a traditional relational database. MongoDB is a cross platform document oriented database, which means that instead of organising data in tables with a fixed schema it organises data in collections that house JSON like documents with a variable set of properties. Information that composes individual digital objects is heterogeneous in nature, with fixed administrative properties that must exist alongside content specific properties that may only be relevant for that particular digital object. In this scenario, the flexible schema offered by MongoDB can be an advantage since it allows the representation of complex and heterogeneous documents within a single collection. On the other hand, MongoDB is usually run as a standalone application which in this case means that those who choose to use the reference implementation of the personal digital repository will need to install and manage an additional piece of software. This issue does not arise in case the reference implementation of the personal digital repository is used in a SaaS scenario, where such responsibilities would be transferred to the service provider. Regardless of the deployment scenario, it should be noted that since this is the storage type that houses the personal digital repository's administrative data it cannot be directly exchanged after deployment. As for the personal digital repository's shared context, the semantic nature of the data that it will need to be stored provides a strong incentive to use a triple or quad store instead of a traditional relational databases. The reference implementation of the personal digital repository comes with support for two quad store providers, one based on Neo4j [126] graph database and the other on OrientDB [127] graph database. Both can be used as embeddable databases managed directly by the personal digital repository's web application, a trait that reduces the number of individual components that need to be explicitly installed and managed by the repository owner, should he chooses to use them. On the other hand both providers are general use graph databases, not triple or quad stores. Though the basic principle is the same (i.e. at its core both store the representation of graphs) the specific rules and query languages which are expected to be present in quad stores are not available out of the box from these providers. To mitigate this issue there is the need to adopt an abstraction layer that enables generic graph databases to function as quad stores and be queried using SPARQL instead of their own graph query language. The required abstraction layer is provided by the Thinkerpop Blueprints generic graph API [128], which allows different underlying graph databases to be abstracted and exposed using the OpenRDF Storage and Inference Layer (SAIL) interface. Internally the core repository service provides and abstract quad storage service that is responsible for instantiating and initialising the chosen storage engine. This is done to ensure that only one of available storage engines is used at any given time. It also allows to avoid a direct dependency on specific storage engines that might arise in modules that provide services other than semantic storage, since they should always invoke the generic repository storage service instead of one provided by a specific implementation tied to one of the storage engines.

Differentiated treatment of content is provided by a set of specialised content modules. For the reference implementation of the personal digital repository was adopted a strategy that call for the use of repository modules to support each one of the available content collection tools. This is reflected by the implementation of modules to deal with browser history entries, text messages from mobile devices, social media posts and chat messages, with each module implementing operations from the "/dataManagement", "/content" and "/search" groups. The "*contentHandler*" parameter is used by the core repository module to determine to which of

the available content modules incoming requests should be forwarded. Requests that arrived without the "*contentHandler*" parameter are treated as generic requests and assigned to the core repository module itself. This results in only minimal information extraction when a request to store content arrives, or the generation of a generic visualisation composed of administrative metadata alongside a link to the content. In the reference implementation of the personal digital repository the definition of what constitutes an acceptable ingestion strategy is part of the content modules responsibilities. For each module its corresponding content collection tool will need to follow its set strategy, which can vary accordingly to the content type. For some types of contents this means that a single request to store content will correspond directly to one digital object, while for others a single request might carry with it multiple digital objects, be them of the same type or of distinct types. Regarding the "*showGraph*" operation of the "/content" group the reference implementation of the personal digital repository offers a service that given a starting digital object generates a graph representation of other objects and properties, up to a configurable node distance (which is by default up to 3 nodes). The included graph service is based on graphviz [129], though alternative services could be provided by other repository modules. Whenever the submitted content is not addressed to one of the existing content modules, it is treated as if it were generic content. The core repository module assumes the responsibility to ensure that it is processed and stored, going so far as to attempt to extract metadata from the content by feeding it to Apache Tika, if the submitted content is file based. Besides registering its ingestion in the shared context, no other action is performed with the extracted metadata, being that in the future, the extracted metadata should be passed to an internal module for mapping with the base repository ontology.

As an additional remark, this reference implementation was not designed to be used directly, but mainly to serve as a test bed for the storage, shared context and the content collection tools. This means that although there is an user interface in place, it is not intended to serve as a daily driver for direct interactions between the repository owner and the personal digital repository other than administrative interactions. Each content module, as part of its functions can provide visualisations to be used by the end users when interacting with gathered content, while the repository itself only serves as a fallback for when such interfaces are not available are not available from the modules themselves. Given that the reference implementation primary goal was to serve as a test for the content collection agents, user interface for content navigation has yet to be fully implemented, and until them interactions with the repository are expected to be done primarily through the content collection agents. In the future, when the user interface is implemented it will enable seamless navigation in the content gathered through the personal digital repository.

5.2 Content Collection Tools

Content collection tools created to support the reference implementation of the personal digital repository are focused on non traditional content. This choice was made to showcase how much information can be gleaned from often overlooked sources and how such information can contribute for the creation of the repository's shared context. Content collection tools come in a variety of formats, ranging from plugins for existing applications, to web applications to native applications. The choice of format is done in order to attempt to minimise the impact that the collection tool would have in the repository owner's habits. The rational behind this being that if it can be integrated in an application that the repository owner already uses,

the chances of actually be adopted and used increase. Additionally, in order to further spur its adoption, content collection tools should provide a convenience service to their users. This service should, if possible, be provided even without relying on the existence of a personal digital repository, though it may be enhanced if the collection tool is able to connect with a personal digital repository.

Browser History Collection Tool

The browser history collection tool was devised as an extension for the Google Chrome browser [130]. This tool provides an alternate implementation of the browser history service that relies on additional elements to improve recall efforts. Instead of relying only on the page address or its title this extension gathers semantic information (in RDFa format) from the visited pages and captures a thumbnail image of the page when it finishes loading. These additional elements are initially stored locally and can be queried by the user when it attempts to locate a previously visited page. When compared with the browser history that comes with the browser this approach allows the user to explore additional options when attempting to locate a previously visited page, such as searching for specific topics or authors as long as the visited page is annotated with such information. Furthermore it can be used as exploration tool for the semantic web, since it notifies the user to the presence of semantic elements in the visited pages.

The connection with a personal digital repository is presented as a convenience service that allows the browsing history to be transferred across devices and archived. This strategy helps to skirt the limits imposed by browsers to how many and for how long history entries are kept. When searching, entries transferred to the personal digital repository can be included in the search results, though this comes with the downside of increasing the query time since the personal digital repository needs to be contacted and the result set needs to be merged. Content transferred from and to the personal digital repository includes not only the history entry itself but also any additional semantic information and the thumbnail for the visited page. Synchronisation between the collection tool and the personal digital repository can be either complete, where the entire browsing history available to the content collection tool is sent to the personal digital repository or incremental, where only new entries gathered since the last synchronisation are sent to the personal digital repository.

Interactions between the browser extension and the personal digital repository are mediated by a content module that is responsible for storing and retrieving the gathered information. The browser extension does not submit for storage in the personal digital repository individual entries but instead groups of entries in JSON format (with the thumbnail included as base64 encoded content), up to a maximum size of 2Mb. In the personal digital repository side the corresponding content module is responsible for processing each individual history entry and placing it both on metadata storage as well as in the shared context. The content module takes advantage of the fact that the underlying semantic storage is a quad storage and uses it to create a named sub graph with semantic information specific for each entry. This strategy avoids the need to reify incoming collected statements which are made about the content of the pages and may not apply to the shared context in its entirety. The content module also handles search requests that come from the browser extension. Search is performed over the content as stored in the shared context, and as such is SPARQL based.

SMS Collection Tool

Another developed content collection tool was aimed at gathering the text messages residing in Android mobile devices [131]. For the end user the tool provides a backup service that can be used to export existing SMS messages, call logs or the contact list to an external file, so that they can be transferred to and eventually restored in a different device. The main difference between this and other available backup and restore solutions is that this one attempts to augment the collected information with contextual location information. This means that when a message or call is made or received, or a contact created the application attempts to record the device's current location, in the form of coordinates, GSM Cell-ID/Area Code and address. It should be noted that the address is only resolved from the gathered coordinates if there is an available data connection, and that the precision of the coordinates gathered will depend on the available location providers. Though not present in the current version of the application, location data could be used by the device owner to see (roughly) where each one of those events happened by for instance placing pins in a map.

The connection with the personal digital repository is presented to users as a convenience service that allows them to use the "cloud" as a storage medium for their backups. In exchange the personal digital repository gains access to the device's contacts, to the messages and logs, and to the location data. This content can be used by the content module in the repository side to create a basic representation of individuals that may be acquaintances of the repository owner, to connect them with contact points (primarily phone numbers) and to associate geographic information to message related events, thus enriching the contextual information available in the shared repository. Additionally, this content collection tool recognises that its users may not want all the messages to be gathered and preserved (for instance due to being potentially compromising or sensitive in nature). As such and although it runs afoul of the "store everything" mantra that guides the personal digital repository it allows its users to mark individual messages to be excluded from backup procedures, so that the copy over which the repository owner has control only remains in that particular device where it currently resides. Furthermore, it also recognises that always carrying a complete history of all messages ever sent or received may not be practical. Thus, when retrieving and restoring messages from the personal digital repository it allows its users to establish an (optional) cut-off date, with messages older than the specified date not being retrieved from the personal digital repository to the user's device.

Social Media Collection Tool

The Social Media Collection tool is aimed at gathering content from social media accounts [132]. For the end user the tool provides a service that creates an offline backup representation of the content present in social media accounts. Given that the tool is capable of interacting with multiple types of social media accounts, one of the possible uses for the created backups is to be used to recreate an account either in a different service provider, or in the same one under a different account if the original version is taken down, becomes unavailable or its owner lost control over it. The content collection tool is presented as a web application that is capable to interact with several social media platforms (such as Facebook, Linkedin or Twitter) through their respective APIs. After choosing one of the supported social media platforms and authenticate with it, the user is able to extract his content from the platform, being offered a version that can be stored in the user's local device. The offline version of

the account is composed by sets of RDF files (with the generated sets being different for each platform) to describe the content's properties and extracted multimedia content (for instance images or videos). The RDF files are not designed to be viewed directly, but instead to be processed and used by the content collection tool when restoring or migrating an account or by the personal digital repository. On the other hand, multimedia content can be seen directly (albeit without its accompanying context that is located in the structure RDF files). After obtaining the content its multimedia part can then be uploaded to another service (in case Google's Picasa) in order to create alternative replicas to the ones that reside in social media platforms. The content as a whole can also be uploaded to the personal digital repository.

On the personal digital repository side incoming content is handled by a dedicated module. This module is responsible for processing the uploaded set of RDF descriptions that represent the social media content. A digital object is created for each recognised record (for instance posts, messages or contact information) that is not yet in the personal digital repository, being that potentially duplicated content is identified by the url that provides a link back to it in its original context. In addition to content produced by the repository owner the gathering tool is able to collect contact and profile information about the repository owner himself an those with whom he interacts. This additional information is used by the personal digital repository to complement its shared context, particularly the social relations part. The content module is also responsible for handling the retrieval of content in a raw format, suitable for use by the content collection tool in its service of account restoration or replication. To this end, the set of received RDF files is itself considered to be content and stored in the personal digital repository so that it can be recalled by the content collection tool if the need arises.

In its current state, the social media content collection tool is presented as a single user web application. This approach has the disadvantage to add another software piece to the ones that will need to be managed by the user who choose to host their own personal digital repository. Alternative approaches to this approach can range from merging this collection tool into the personal digital repository itself (for self-hosted repositories), to providing it as complementary service when deploying personal digital repositories as services themselves or even to modify the application to include user accounts and to keep the necessary information in it to be able to interact with multiple personal digital repositories, deploying it afterwards as a standard web application. Furthermore, the collection tool's capability to restore a social media account is limited by the services exposed by the API of each of the social media platforms supported. Though it is possible to access personal profile information, most platforms do not allow external applications to change it. Another example is that while it is possible to gather the replies that other individuals left to content created by the target account owner, it is not possible to force the platform to directly associate those replies with the individuals who made them when restoring an account (as that would effectively amount to allow external applications to impersonate other individuals). Workarounds for these issues include leaving the original content (such as profile information) as private messages and relying on the target account owner to perform part of the restore process manually, or simulating replies which though still effectively made by the target account owner include the information of who originally replied as well as when that reply was originally created, effectively marking the restored version as an artificial construct. Given these limitations, restored accounts will never be perfect replicas of the source accounts.

Chat Collection Tool

The Chat Collection tool is aimed at gathering online conversation messages from chat applications. The tool is presented as a plugin for the Adium instant messaging client [133]. Adium is open source and comes with native support for several instant messaging protocols (such as Jabber, IRC or ICQ). Its modular nature means that supported protocols, as well as other functionalities can be added to the client with the use of plugins, such as the proposed collection tool. For end users the tool provides a backup services that replicates conversation logs to the cloud, which in this case means that they are sent to the personal digital repository system.

In order to collect messages the plugin taps into the common messaging infrastructure and intercepts messages as they are being sent or received. This strategy has a caveat when it comes to outgoing messages. Since outgoing messages are captured at the end of the processing phase (i.e. after being written but before being actually sent) there is the risk that some of them may not be delivered (for instance due to network issues or simply because the recipient has closed his messaging client and the protocol does not deliver "backlogged" messages), though they were nevertheless captured as if successfully sent. Besides the message itself the plugin also attempts to capture the user's current status message. This additional content can provide an indicator of the user's state when he sent the message (for instance if he was available for a conversation or busy) as well as the user's mood, if the message contains user defined text, or even the music he was listening at the time it was having the conversation with one of its contacts. Detecting if the status message is a simple user defined string or represents the music the user was listening depends on what other plugins are installed in the chat client as well as on the audio player being used. Adium has native integration with iTunes [134], which when used will format the status message in a way that makes it recognisable as representing content that comes from audio player, though what is available to be retrieved are the component tokens used to style the status message, not the values of the tokens themselves. Other plugins provide integration with social media service last.fm [135] and format the status message with a service specific string that, much like the native integration with the iTunes is then replaced with a message. The chat collection tool plugin takes advantage of this last method and when detects that the status message is composed by a string that is used to display information from the last.fm service makes its own call to the service's API to retrieve information about what was being heard, thus sidestepping the issue of not being able to resolve directly the tokens that compose the specially formatted status message string. Retrieved information is then appended to the data that will be sent to the personal digital repository. By default collected messages are sent to the repository addressed to the "simpleChatLog" content module as soon as they are sent or received, though if there is a communication failure between the plugin and the personal digital repository processed messages are stored in JSON based log file and replayed at a later opportunity (i.e. when the chat client is restarted).

On the personal digital repository side incoming content is handled by a specialised content module. In addition to storing received messages, the module is also responsible to create conversation (i.e. a collection of messages that are related to each other). A conversation is created whenever the content received does not include an conversation identifier. The conversation identifier is only created after receiving the first message and is returned to the collection tool as part of the acknowledgement that lets it know that a message was successfully stored in the personal digital repository. The module also performs a basic form of content

analysis on message's text and attempts, to the best of his ability, to detect whenever an url has been mentioned. This is used to connect a message sent (or received) event with a given url. Depending on the direction of the message, it can them be used to state in the personal digital repository shared context that a given url was suggested to the repository owner, or suggested by the repository owner.

5.3 Example Of Collected Information

Content gathered by the collection tools has a dual representation in the personal digital repository, one in the digital object catalogue and another in the shared context. The first one acts as an administrative record for the content and in the reference implementation is represented in a JSON structure. While the core administrative metadata is stored in a normalised way, storage of object specific metadata is not subjected to this restriction. It is the responsibility of the content module that initially received it to store in the digital object catalogue any additional metadata that pertains only to that digital object with the caveat that the representation of that data is both content and module dependent. In the reference implementation, object specific metadata being stored at the same level as the administrative metadata, though it should have been stored underneath a dedicate "metadata" field as defined in the architecture. It should be noted that relations between objects are represented in the shared context, not in the digital object catalogue. This means that object bound metadata is stored in the digital object catalogue in a intermediate format (as close as the original format as possible) that needs to be interpreted and normalised by a content module before being integrated in the shared context. This is done in order to preserve the original metadata, so that it is possible to rebuild the shared context from the stored objects if the need arises. An example of the administrative metadata representation of a digital object (in the case an SMS message) can be seen in Figure 5.1. Notable points in the example are the inclusion of an extra field *version*, that comes from the version mechanisms of the web framework itself, the use of *contentFormat* value that stems from the MIME personal tree (prs prefix) intended for experimental uses, as well as the duplication of the message's content in the *description* field to support search.

The base representation of a digital object in the shared context is normalised by the content module that first received it to conform with the CIDOC/CRM ontology and supporting extensions proposed in the previous chapter. The representation will necessarily change over time with the establishment of additional relations that is brought either by the addition of new digital object or by the reinterpretation of existing ones by other content modules. To illustrate this consider the scenario of an instant message exchange between the repository owner and one of its friends. The exchange is captured by the chat collection tool installed in the repository owner's chat client, which streams sent and received messages in near real time to the personal digital repository, being that one of those messages actually contains an url with a suggestion of a web page to visit. The repository owner does not follow the suggestion right away but instead only in the next day, and does it so in a browser that has the browser history content collection tool installed, which means that the visit is recorded and eventually sent to the personal digital repository as well. The representation of this scenario in the shared context of the personal digital repository can be divided in two parts. The first one that deals with content itself and a second one that deals with internal records generated by the personal digital repository.

```
{
  "_id" : ObjectId("560d53e706926aed02a6a566"),
  "version" : 0,
  "ingestionDate" : ISODate("2015-01-01T15:30:35Z"),
  "modificationDate" : ISODate("2015-01-01T15:30:35Z"),
  "initialHandler" : "simpleChatLog",
  "contentFormat" : "text/prs.ChatMessage",
  "contentType" : "text/prs.ChatMessage",
  "revision" : 0,
  "checksum" : "affe25d413aca5bbf64136a4b385ef343d6211d3",
  "checksumAlgorithm" : "SHA-1",
  "contentStorageType" : "inline",
  "content" : "Check out www.example.com",
  "description" : "Check out www.example.com",
  "metadata" : {
    "sender" : "john@chatservice.com",
    "receiver" : "mary@chatservice.com",
    "date" : ISODate("2015-01-01T15:30:01Z"),
    "conversation" : ObjectId("560d53e706926aed02a6a564")
  }
}
```


Figure 5.1: Administrative metadata (as represented in MongoDB)

The first part is shown in Figure 5.2 shows the representation of user-driven events and content in the shared context of the personal digital repository. Entities are represented by blue and yellow boxes, with its class on the blue part and the value in the yellow one while properties are represented by arrows. Identifiers assigned to entity instances are automatically generated by the personal digital repository and assigned to digital objects when its administrative metadata is stored, or directly requested to the underlying database in case of entities that need such an identifier but are derived from the properties of the digital objects themselves; "holder" entities are shown with the value of their properties directly on the yellow box. There are two distinct participants in this scenario, the repository owner and a friend, and each is represented by a different entity type that reflects their roles in the personal digital repository. So, Mary the repository owner is represented by a EPDR4 Repository Owner entity while John is represented as a more generic P21 Person entity. It should be noted that their names are separate entities, represented in through the use of E82 Actor Appellation entities. Each of the participants has access to a contact point instance, that is used when attempting to communicate with each other. The attempts are represent as instances of EPDR25 Chat Communication Attempts which act as hubs, establishing the relation between the actual conveyed message, the participants and temporal instant when the communication occurred. The temporal dimension is represented through instances of the E52 Time-span entity. If their value is coarse enough, conceptually these instances could be shared between events to implicitly establish a loose temporal connection, tough care should be taken to interpret such connections as anything more than temporal coincidence without any additional data. As for the messages themselves they are represented by EPDR17 Chat Message instances, with their actual textual content associated to it as an informal and unstructured description through an P3 has note property. This is done to allow the textual content of the message to remain in the shared context, despite the shared context being more geared to describe the events and properties that surround the message, not the the content itself. On the other hand, information extracted from the textual content itself, such as the presence of an URL should, if possible, be part of the shared context. In this scenario that situation is represented by establishing a connection between one of the messages and an instance of a EPDR67 Web

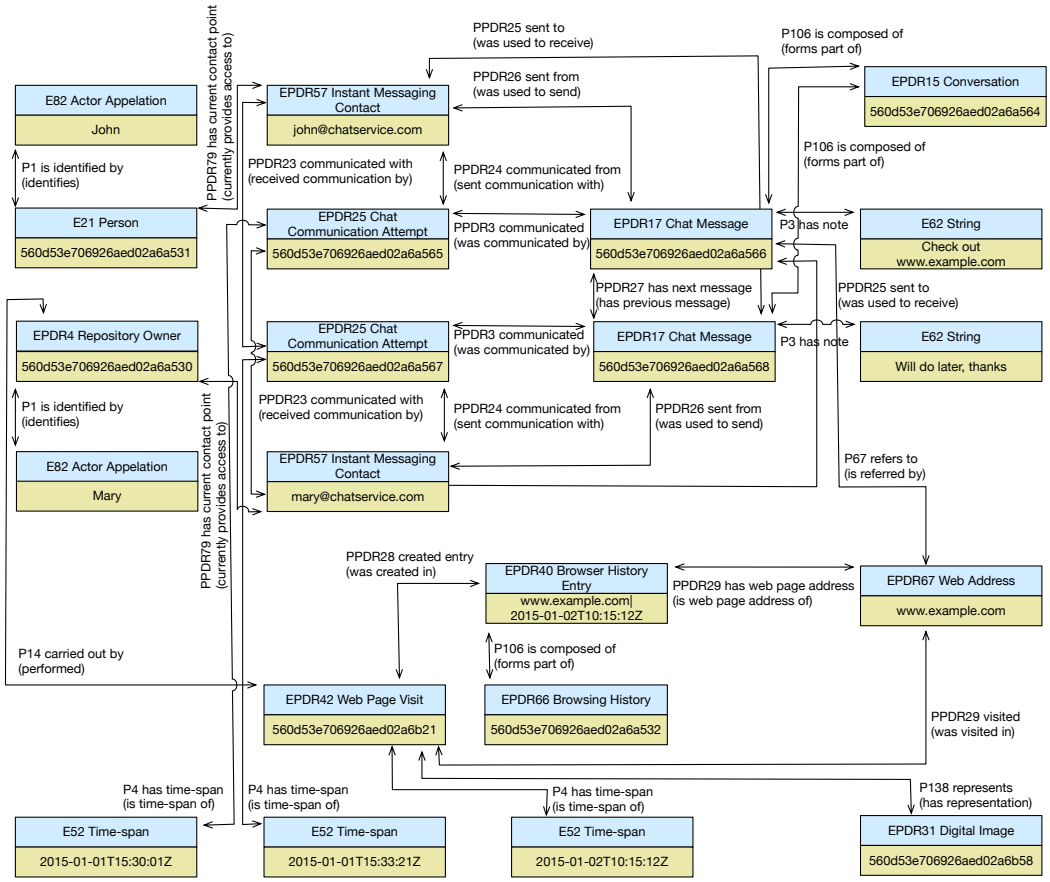


Figure 5.2: Representation of a message exchange (Part 1)

Address, that if not yet present in the shared context needs to be created. Regarding the repository owner's later visit to the web page, it should be noted that the screenshot taken by collection tool is going to be associated with that particular visit, as it offered a depiction of the page at the time of that visit. Subsequent visits may result in different representations (and thus visually different screenshots).

The second part of this scenario, illustrated in Figure 5.3 deals with the events taken by the repository itself as it ingests a digital object from the previously described exchange. From the personal digital repository's perspective (which is conveyed by the plugin that handled the content processing), there are a number of internal events triggered by ingesting new content. In this scenario the first event that emerges is the ingestion itself, represented as an instance of EPDR7 Ingestion event. In addition to marking the temporal aspect of when the content was ingested, it is also during this event that content is marked as currently belonging to the repository owner and that it is specified to which repository the content was ingested (as the shared context may have information about the existence of other personal digital repositories). Following the ingestion, the next event is to enact over new content any standard policies such as marking as being private or in alternative to apply custom policies specified by the current content processing plugin. The final step in this scenario arises from how the content plugin handles chat messages, as it establishes that every received chat message should be part of a conversation. This conversation is either specified by the content collection agent, or if

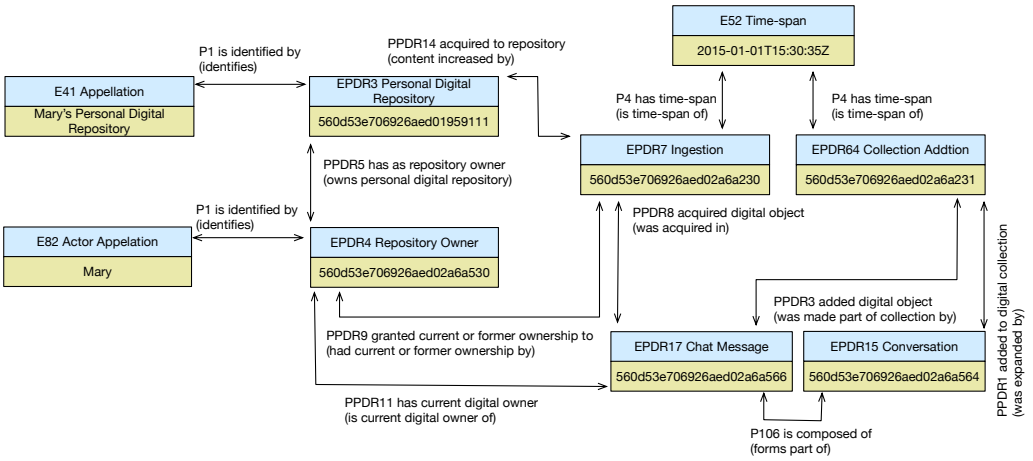


Figure 5.3: Representation of a message exchange (Part 2)

missing created on the fly and its identifier returned to the content collection agent for future use. Though the creation part is omitted for simplicity sake, adding content to a collection is also an event which needs to be represented. In addition to the temporal dimension the event is also established a connection with the added content and the updated collection. In a final remark, though each of the depicted events are independent, the processing plugin uses the same date for all of them, which results in a graphical representation that ties them to a common time-span node.

While not directly encouraged, the repository owner can submit content directly to the personal digital repository by addressing a request to the submission URL (either through a submission form or as a raw request). To illustrate this, consider the scenario where the repository owner submits a copy of this thesis in PDF format for storage in the personal digital repository without specifying any content handler parameter. Given that no content handler is specified in the request and that no content handler is registered to deal specifically with content in PDF format (as the developed content modules are geared towards supporting the previously presented content collection agents) the core repository module would have to act as an handler of last resort. As part of the process, the submitted content is passed to Apache Tika for metadata extraction. In the implemented version, the core repository module limits itself to gather DC elements from all the possible metadata returned by Apache Tika, as these can also be mapped onto the proposed ontology, which results in the administrative metadata representation seen in Figure 5.4. Notable departures from the previous example include the content storage strategy, whose key fields, "*contentStorageType*" and "*content*" now reflect the fact that the content will be stored in the local file system by the core repository module and the inclusion of the "*name*" field with the original file name. Additionally, the "*description*" field is missing, though it could be used to store, for instance, the textual content of the submitted document as extracted by Apache Tika in order to make it searchable.

Even without a specialised handler, content still needs to become part of the repository shared context. Common metadata elements, such as the ones that compose DC can be directly mapped to the shared context underlying ontology. Thus metadata elements such as "*created*" are unfolded into derivate instances from the CIDOC/CRM E5 Event entitiy, while

```
{
  "_id" : ObjectId("568be4ca53b37a7d93049762"),
  "version" : 0,
  "ingestionDate" : ISODate("2016-01-05T15:45:25Z"),
  "modificationDate" : ISODate("2016-01-05T15:45:25Z"),
  "initialHandler" : "pdrCoreModule",
  "contentFormat" : "application/pdf",
  "contentType" : "application/pdf",
  "revision" : 0,
  "checksum" : "ac7a675949f71c154b82f5312c4d21ccfc031e8a",
  "checksumAlgorithm" : "SHA-1",
  "contentStorageType" : "internal",
  "name" : "PersonalDigitalRepository.pdf",
  "content" : [
    {
      "module" : "pdrCoreModule".
      "identifier" "/Users/marcopereira/Storage/ac7a675949f71c154b82f5312c4d21ccfc031e8a.pdf"
    }
  ],
  "metadata" : {
    "dcTitle" : "A Cloud Based Semantically Enhanced Personal Digital Repository",
    "dcCreator" : "Marco Pereira",
    "dcSubject" : "Personal Digital Repository, Content Gathering, Semantic",
    "dcTermsCreated" : ISODate("2016-01-05T14:46:33Z"),
    "dcTermsModified" : ISODate("2016-01-05T14:46:33Z"),
  }
}
```


Figure 5.4: Administrative metadata for a pdf file

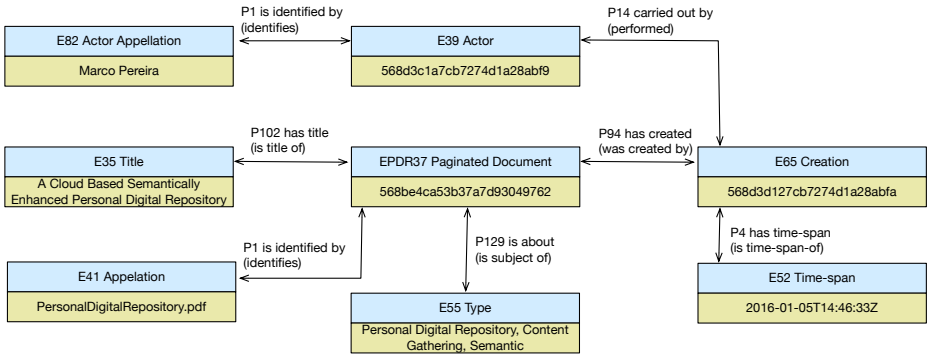


Figure 5.5: Representation of DC metadata from a pdf file

---

others, such as the "*title*" element become additional content identifiers derived from the CIDOC/CRM E41 Appelation entity. Figure 5.5 uses the pdf file from the previous described direct submission scenario to illustrate how the DC metadata extracted by the core repository module is represented in the personal digital repository shared context. It should be noted that while present in the supplied metadata, the "modified" element is not represented in the shared context. This is happens since the corresponding element from CIDOC/CRM, E11 Modification is only applicable to physical objects, since modifications to immaterial objects do not leave behind physical traces instead resulting in the immediate creation of a new one, while the event from the personal digital repository ontology, EPDR5 New Version, is meant to represent when a new version of a given object was generated, not when it was last modified in case of the its initial submission to the personal digital repository. If the content were to be submitted as an update to a version previously submitted to the personal digital repository, then the EPDR5 New Version would be applicable and could use the data extracted from the "modified" element.

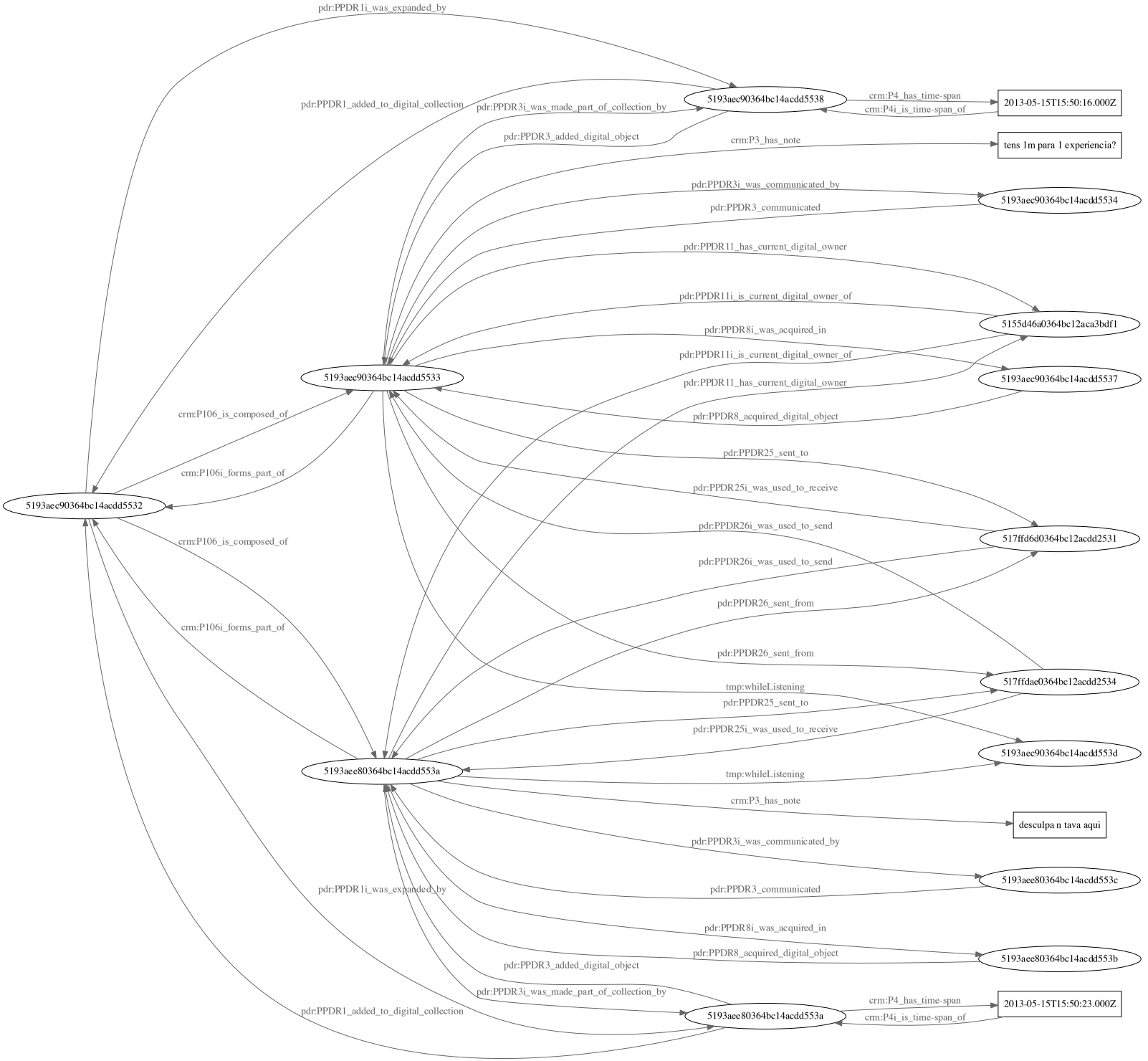
As a standard feature the personal digital repository includes a provision for a "showGraph" operation whose goal is to provide a visual representation of how interconnected a given content piece is. In the reference implementation of the personal digital repository service this

is handled by a graphviz backed service that receives a list of quads (triples with an additional context property) from which to plot a graphical representation, that is returned in the form of an svg file that can either be embedded on a page or returned to the client as-is. It should be noted that it is the responsibility of the content plugin that is handling the request (or the core repository) to generate the list of quads that will be represented. For instance, in the reference implementation of the personal digital repository the module that deals with chat content generates graphs centred around a given conversation, with a maximum depth of 2 (i.e. the graph only contains nodes that were either directly connected to the starting conversation collection or connected to those connected to the starting conversation). This has the side effect of hiding some information (for instance dates) that would come from nodes in the third level. An example of the service's output when applied to a message exchange can be seen in Figure 5.6, though it should be noted that it was generated before the completion of the ontology that currently underpins the personal digital repository and as such has some temporary terms displayed. Additionally, the dot "output" was used instead of the svg, being slightly modified for display purposes on the page, with the complete URLs that identified the properties being replaced by name-spaced versions, as well as the removal of the common URL part ("*http://localhost:8080/personalDigitalRepository/content/showContent/*") of the identifier displayed in the nodes.

5.4 Chapter Conclusions

This chapter provided an overview of the reference implementation of the personal digital repository. The main goals of the reference implementation were to provide a material support for the personal digital repository, particularly to the programming interfaces, services and data structures proposed in the previous chapters, as well as to serve as a test bed to demonstrate how a possible intertwined system of content collection agents focused on unconventional content can contribute to the creation of the repository's shared context.

The creation of a reference implementation was a required step to demonstrate that the concept of a modular personal digital repository could be implemented. As a backend for administrative metadata storage the reference implementation uses MongoDB due to its flexible document oriented model that is flexible enough to merge within a single record (or in MongoDB terms, a single document) fixed metadata (required for the operation of the personal digital repository) and a variable, content dependent metadata. For the shared repository the reference implementation of the personal digital repository provides an embedded triple store backed by graph databases (either OrientDB or Neo4j). The use of triple stores reflects the need to represent structured information whose interconnections, be them internal from a single object or external between multiple objects, and their type can be as important as the data itself and allows a personal digital repository to provide a canvas upon which the relations between apparently disparate digital objects can be pinned, often by way of third party entities already present in the repository. As for the modular architecture, in order to achieve it the reference implementation co-opted the plugin mechanism already present in the Grails web application framework. Ideally repository modules should be isolated in a per user basis and their installation, activation or removal should not interfere with the regular operation of the personal digital repository. Co-opting the Grails plugin system came with the need to compromise this ideal operation mode, as Grails' plugins are required to be present at the very least when the web application is started to be recognised and loaded,


pdr:PPDR1i_was_expanded_by
pdr:PPDR1_added_to_digital_collection
5193aec90364bc14acdd5538
crm:P4_has_time-span
crm:P4i_is_time-span_of
2013-05-15T15:50:16.000Z
pdr:PPDR3i_was_made_part_of_collection_by
pdr:PPDR3_added_digital_object
crm:P3_has_note
tens 1m para 1 experiencia?
pdr:PPDR3i_was_communicated_by
pdr:PPDR3_communicated
5193aec90364bc14acdd5534
pdr:PPDR11_has_current_digital_owner
pdr:PPDR11i_is_current_digital_owner_of
5155d46a0364bc12aca3bdf1
pdr:PPDR8i_was_acquired_in
pdr:PPDR11i_is_current_digital_owner_of
5193aec90364bc14acdd5533
5193aec90364bc14acdd5537
pdr:PPDR11_has_current_digital_owner
pdr:PPDR8_acquired_digital_object
crm:P106_is_composed_of
crm:P106i_forms_part_of
pdr:PPDR25_sent_to
pdr:PPDR25i_was_used_to_receive
5193aec90364bc14acdd5532
pdr:PPDR26i_was_used_to_send
517ffd6d0364bc12acdd2531
crm:P106_is_composed_of
pdr:PPDR26i_was_used_to_send
pdr:PPDR26_sent_from
crm:P106i_forms_part_of
pdr:PPDR26_sent_from
517ffdae0364bc12acdd2534
tmp:whileListening
pdr:PPDR25_sent_to
pdr:PPDR25i_was_used_to_receive
5193aec90364bc14acdd553d
5193aee80364bc14acdd553a
tmp:whileListening
crm:P3_has_note
desculpa n tava aqui
pdr:PPDR3i_was_communicated_by
5193aee80364bc14acdd553c
pdr:PPDR1i_was_expanded_by
pdr:PPDR3_communicated
pdr:PPDR8i_was_acquired_in
pdr:PPDR8_acquired_digital_object
5193aee80364bc14acdd553b
pdr:PPDR3_added_digital_object
pdr:PPDR3i_was_made_part_of_collection_by
crm:P4_has_time-span
2013-05-15T15:50:23.000Z
pdr:PPDR1_added_to_digital_collection
5193aee80364bc14acdd553a
crm:P4i_is_time-span_of

thus requiring a restart of the application server that is disruptive for normal operations, as well as being installed application wide and thus available for every users instead of only for those who actually installed them. On the other hand it should be noted that such tight integration with the framework components alleviate the need to create interfaces for their use, as their are already provided by the framework itself. This means that modules can directly supply controllers, services or views [121], or use those supplied by other modules without the need for an additional compatibility façade maintained by the personal digital repository itself, with the notable exception of those services required to maintain the integrity of the repository (such as with the service that manages the underlying shared context). The previously mentioned compromises affect primarily SaaS multi-tenant deployments of a personal digital repository, with the mandatory application reboot affecting all scenarios. Taking this into account, the integration advantage is enough to warrant that, with mitigation measures in place the Grails plugin system can be used as the basis for the implementation of the required modular architecture of a personal digital repository. A possible mitigation measure that can be put in place for SaaS scenarios is to assign to each personal digital repository its own virtual machine. Despite the predictable increase in the resources that need to be fielded in order to provide the service, this provides a level of isolation that can counteract the previously mentioned disadvantages that came from co-opting the Grails plugin system, being that if all other components are also replicated in the local virtual machine (for instance the management database) it also serves as an additional insurance for the end users of the system that their personal content and information are effectively separated at a logical level from the ones of all other users.

While the reference implementation of the personal digital repository serves as hub for content, it is not capable (nor supposed) to acquire it by itself. To complete the content gathering ecosystem there was also the need to create content collection tools. For demonstration purposes, four content collection tools were created, that focused primarily on non traditional content such as text messages or social media posts. It should be noted that with additional collection tools, or repository modules it would have been possible to support the collection of more traditional content types. Different pieces of traditional personal content, by its own nature, can be assumed to be more closely interconnected with one another, though often such connections are not immediately clear. Indirect connections can be found, or at the very least hinted in the non traditional content pieces, that can serve as the "missing link" between other content pieces. For example a text message (non traditional content) from an acquaintance can suggest the attendance of an activity (that the one suggesting did not attended) from which photos (traditional content) eventually were taken. Without the message to act as a liaison, and without appearing in the photos, the role played by said acquaintance in influencing the repository owner to attend the event could be overlooked or even lost. Non traditional content is thus gathered both by its own sake and by its potential to contribute to the personal digital repository's shared context. The created content collection tools dealt with browser history, SMS and contacts stored in mobile devices, social media content and chat messages. With the exception of the chat collection tool, all of them followed a pattern of offering one or more additional services, some of which interact with the personal digital repository. This strategy serves to promote the use of the tool itself and with it of the personal digital repository and it is not without some degree of risk. Unless there is a readily available deployment of personal digital repositories offered as a service that can be assigned to end users, they may either use only the collection tool for its (from the personal digital repository point of view) secondary functionalities (which are serving as a red herring to attract users).

In spite of this potential risk, the created collection agents can arguably be said to be able to fulfil their role, relaying content to the personal digital repository for future storage and processing. The presented scenario of an indirect interaction between the content gathered by the browser history and chat collection tools is just one of the possible interactions, with interactions of same type being likely for instance if collected social media posts are analysed in the same way.

Despite being successful as test platform, the proposed reference implementation of the personal digital repository can not be considered to be complete. It lacks a compelling user interface, effectively relying on the collection agents as primary means of interaction, while at the same time not being simple for individual users to pickup, install and maintain, thus effectively sharing some of the same flaws that were identified in traditional digital repositories in Chapter 2. Nevertheless given that its aim is to serve as a proof of concept, it can be considered a success in this regard, and these issues can be corrected in future versions, more oriented towards the end user's needs.

Chapter 6

Conclusion

As more and more emphasis is placed on digital interactions, it becomes unavoidable that some of the objects that compose our "external memory" become themselves digital incorporeal entities. After all, we're not keen to remember the support where it resides but the content it provides access to. Progress made them simple to create, produce and manipulate, to the point where shared information goes from the realm of the critically important to become riddled with seemingly trivial affairs. The simplicity to create, access, replicate, modify or share some of these information pieces make them appear nearly disposable. They are created, serve their purpose and them can apparently be safely forgotten. Yet, like the pieces of pottery found in archaeological sites they can have lasting value. These pieces of information are part of our cultural heritage for the future and represent a window to our current way of life, thought processes, worries and interests. On a more personal level, as time progresses, those seemingly insignificant pieces of our past can grow in importance for ourselves and acquire deeper meanings. Their creation helped shaped who we are now, yet at the time of their creation might have been casually disposed without a second thought.

The research done through the course of the doctoral programme that culminated in this thesis attempted to deal with the issues that can arise from the introduction of a digital repository in a personal scenario. A personal digital repository acts as a hub where personal content of any kind can be placed, and where the relations between apparently disparate content pieces can be highlighted. There are several technological solutions to deal with content management, ranging from complete digital repository packages to models that act as guidelines for the functions that should be available from said digital repositories to models for how the content's metadata should be represented. As was seen in Chapter 2, the existing models and solutions are, in their majority, geared towards institutional scenarios. Those scenarios are usually characterised by the existence of a community with shared interests gathered around a repository focused around a specific subject with well defined content collection guidelines and trained personnel that is able to enforce those guidelines, as well as any other policy deemed necessary. Not all of these characteristics are present in personal scenarios, where the community with shared interests is replaced by a single individual, whose shifting interests will cover multiple subjects preventing the creation of an homogeneous and focused repository and where there can be no assumptions regarding how well trained the user will be or about its willingness to continuously feed content into the repository system for future use. This is an additional aspect of personal scenarios, that need to consider not only the technological solutions but also its own users behaviours and motivations. Protracted

explicit interactions with a digital repository, like those required to submit content to existing software systems that were adapted for personal use can lead to user exhaustion and eventual abandon of the platform. This issue is further compounded by the usually relaxed attitude that the majority of individuals has regarding their digital content, that while recognising that their content may be at risk, only adopt mitigation measures when they are personally struck with some kind of misfortune that nearly deprived them of their digital estate, and even then only while they remember said misfortune. it should also be noted that it is difficult (if not impossible) for individuals to accurately assess the importance of each of their pieces of content. A seemingly innocuous and disposable text message could prove to be the starting point for an event chain with life changing impact, while a long, carefully written essay upon which the future of the individual appears to depend may reveal itself as only a footnote in its life. The point here being that non traditional content, that may not even have a "corporeal" manifestation in the form of a traditional file that can be explicitly submitted to a traditional repository can contain as much (or even more) information that will prove to be important to the repository owner in the future, be it for the information itself or by serving as a recollection trigger. These issues guided the research through its three major objectives, the pursuit of each yield the following contributions:

- The first objective was the development of novel approaches for content collection from individuals. The contribution of this work in this regard comes from the adoption of a systems of content collection agents that attempt to be as inconspicuous as possible, preferring working in tandem with existing software to capture content as close to the source as possible, while at the same time provide additional complementary services to spur the adoption of the collection agents, and with it of the personal digital repository. The end result of this is an intertwined system, where the majority of the content present in a personal digital repository comes from the content collection agents at large (notwithstanding direct contributions from the repository owner). Without content collection agents, the personal digital repository will not have access to the content it needs to perform its mission, and without the personal digital repository the complementary services that can be provided by the content gathering agents become limited. It should be noted that in this model, users must opt to use the personal digital repository, but initially do it so under the guise of accessing enhanced versions of services provided by the content gathering agents. Furthermore, the repository itself is built to be modular, with content modules assigned to process the content gathered by specific collection agents. This allows new content collection agents, that deal with different content types (or the same ones but in a different way) to be created and have support from the repository itself by recommending to the repository owner the installation of its associated content modules.

- The second one was the development of a flexible model for the integration of disparate content. The contribution of this work in this regard comes from form chosen to represent digital objects. Digital objects have a dual representation. The first representation is primarily for repository management purposes that stores administrative metadata, as well as "raw" object metadata. The purpose of the "raw" object metadata is to allow the reconstruction of the second representation if the need arises, and to allow the object the creation of a compact metadata package that can be manipulated as a whole. The second representation is a semantic representation whose purpose is to convey the relations between different objects as established by shared properties, temporal sequence,

social relations or simply by user defined categorisation and organisation. It relies on a proposed ontology that serves as the baseline to which the multiple metadata sets extracted from the digital objects proper. The proposed ontology has been designed to be extended in order to be able to deal with new content types and metadata elements that may arise during the repository owner life. This, in conjunction with the "raw" object metadata stored in the administrative representation allows the representation of content to evolve, as content already gathered can theoretically be re-evaluated by new content modules that bring with them updated ontologies and properties. This in turn allows the repository as a whole to evolve along the shifting interests of the repository owner, that are also expected to evolve over time.

- The third objective was the incorporation of non traditional content into the personal digital repository. The contribution of this work in that regard comes from its adoption of non traditional content as a primary target for content collection. While individually a single content piece of non traditional content may be labelled as mundane and be less interesting than their more traditional, often file based, content heavy brethren, in numbers they can provide pointers about an individual's thoughts, interests and routines. Furthermore, non traditional content can act as additional an additional metadata source for more traditional content, and the strategies required to collect and store them have helped to tip the personal digital repository in the direction of being more a meta-repository that sends its users towards the content in its original context that a traditional repository that attempts to own the content and strives to become the authoritative source for it. Nevertheless it should be noted that the personal digital repository can deal with traditional content, though it requires the development of additional content modules to take complete advantage of it.

The development of a model and architecture for a personal digital repository comes with its own set of challenges. Content in personal scenarios has the tendency of being scattered through multiple devices and platforms, which according to the research objectives needed to be addressed individually. To further compound this, each platform and device can potentially hold multiple content types, each one of it with a different degree of risk associated with it and different constraints regarding its right and ownership. These issues needed to be reflected both in the collection strategies and in the personal digital repository itself. Regarding the collection strategy, it became clear that it is impossible to develop a single unified strategy that can fit all content types that can be produced in a personal scenario. Instead, one needs to recognise where in its production cycle one should intervene to gather it, and how the intervention can contribute to keep users engaged with the personal digital repository. Given the assumption that neither the personal digital repository nor the content owner can accurately estimate how important a given content piece will eventually become, the adopted approach was to attempt to collect everything, preferably at the moment the content becomes available (i.e. it is shared, posted or sent in case of non traditional content, or when it is saved for more traditional content). In addition to a potential flood of information, applying this strategy as-is can lead to transference to the personal digital repository of content that also exists elsewhere and that is not directly at risk (i.e. is not application or device bound). To counteract this, the adopted collection strategy privileges gathering metadata instead of the content itself whenever the content is not deemed to be at immediate risk. This is done so that the collected information can initially contribute to a shared context maintained by the

personal digital repository, deferring the actual collection of the content to a later date if the content is deemed to be at risk. This results in a paradigm shift, where the personal digital repository behaves less as a content repository and more like a meta-repository concerned with content organisation that can later be used to lead its owner back to the content itself in its original environment, if said environment is still available. To collect content as early as possible and with minimal interference, the chosen strategy was the creation of a series of content collected agents that offer ancillary (from the content gathering point of view) services to its users in order to promote its use and offer adopters something that can give them an immediate benefit. Provided services can in turn be extended with the use of the personal digital repository, initially framing it as a data staging ground. It is only if the users opt to use the extended services that content (or metadata) is actually sent to the personal digital repository. This creates an opt-in system under the control of the user and from which he can back out at any moment.

The use of multiple content collection agents had a direct repercussions in the proposed architecture of the personal digital repository. It highlighted the need for it to be as modular as possible, allowing for the inclusion of entry points tailored for each content collection agent, and its reliance on the content collection agents for content highlights the need for constant connection. Thus the chosen model for a personal digital repository was that of a modular web application, with two types of modules: a core module and an arbitrary number of auxiliary modules. By being thought from the ground up as a web application it incorporates in its design the need for constant connection in order to be truly useful. This allowed the use of standard web protocols and models (HTTP and a REST-like approach) for the development of the repository and its modules, while at the same time it enshrines that the standard interactions between users and the personal digital repository occur through a web browser, an application that should be familiar to most users. Regarding its modules, the core module serves as the foundation upon which the personal digital repository itself will be established. This module is responsible for providing functionalities that can not or should not be offloaded to user definable modules, such as the definition of communication interfaces (be them internal or external), basic digital object definition and management, security and module management. Repository modules are tasked with providing support functionalities, such as processing requests from content collection agents or providing additional storage options for content placement. The external communication interfaces defined are based on a REST-like approach, with a set of interface URLs that form the basis for most of available functionality of the personal digital repository. The core module provides a skeleton implementation of each operation defined under each operation group (or a full implementation if dealing with core functionalities) though it expects operations to be available from specialised content modules. Thus in effect most of the times the core module ends up acting as a pass through proxy for operations performed by other content modules, that can be requested directly by the calling applications (for instance, content collection agents). Repository modules are not limited to the operations exposed by the core module and can expose additional ones, with the caveat that they should do it in their own operation group. The core module is also responsible for applying basic security policies (i.e. to ensure that the calling client is in fact entitled to invoke a given operation, but not if it has access to the requested content as that is left to discretion of the responding content module). The security element that is composed primarily by an authentication requirement that should ensure that clients (i.e. content gathering agents or a web browser) is authorised to perform a given task. Though it would be possible to apply some form of encryption to collected content, the same would not

be feasible for the personal digital repository's shared context. As such the security measures taken by the personal digital repository are more akin to a common lock in that they are there to ensure that those who might be tempted to peek if the repository was unprotected will be discouraged from doing it, but will just be an hindrance for an determined attacker (such as hardened state sponsored hackers).

Metadata representation in the personal digital repository is split into two complementary halves: the first one is the digital object metadata catalogue and the second is the shared context. While both store information about the content gathered by the content collection tools, their internal logic and operation mode are completely different. The metadata catalogue stores information about the digital objects in an atomic package (i.e. in a form that can be manipulated as a whole). The information it carries is divided into a fixed administrative part, that is composed of fields primarily relevant for the internal working of the personal digital repository itself such as identifiers, and by a variable part that is composed by a free form content dependent part. Both parts must be filled by the content module that processed the content when of its ingestion. The metadata catalogue was also used to store additional configuration information and thus is a service that is provided by the core repository. On the other hand, the shared context breaks down digital objects into its metadata components and uses them to establish relations with other objects. This provides a semantic network that can be followed around in order to hop from object to another, limited only by the kinds of relations that the different types of content modules are able to establish with each other. The rules that govern how to represent objects in the shared context are governed by an ontology. This ontology is an extension of the CIDOC/CRM model, adapted to the needs of a personal digital repository. The CIDOC/CRM model was chosen to serve as the starting point for the personal digital repository ontology due to its emphasis on temporal information (that can constitute a fall-back resource when establishing connections between digital objects), availability of extension points and due to having been designed to contain within itself user defined hierarchies. Since the vanilla model did not provide the concepts needed to describe digital object (as it is focused on describing information regarding physical objects, with some support for their underlying concepts and uses) there was the need to extend the CIDOC/CRM model. This extension provided it with concepts it needed to represent purely digital objects, added events for digital activities (both those related to collected digital objects as well as those related with the normal operation of the personal digital repository) and expanded the model's representation of social relations (a key point to represent content that comes from social networks). The strategy adopted for these extensions was one of explicit definition, where instead of creating an high level concept and complement it with a classification from a controlled vocabulary using the model's type system, the added concepts definition includes its classification. This results in a more verbose (i.e. with more concepts needed) and less flexible system (since new concepts need to be explicitly introduced as opposed to the introduction a neutral type identifier), but it releases the type system for future use by the repository owner. This allows owners to create an additional personal classification overlay above the formal one in use by the personal digital repository itself. Arguably, even the proposed extensions will not be enough to represent all the types of digital content that can be accrued thorough the lifespan of a personal digital repository. Both content types and the repository owner's interest levels evolve over time, which means that existing content may at some point have to be reassessed and new types of content (or simply new detailed properties) may emerge and need to be represented. To address this issue the proposed extension itself is thought to be extensible, with entry points being left for the introduction of new digital object types or

properties. These should come as part of the content modules installed in the personal digital repository, and it is their responsibility to ensure that they are exposed to the outside world, so that their definitions can be reused in other modules.

All of the previously described components came together in a reference implementation of the personal digital repository. This reference implementation comes in the form of a Grails based web application that uses the framework support for plugins as a means to implement its modular architecture. Relying on the underlying framework for the modules implementation comes with the downside that they become globally available and require an application restart in order to become active. Digital object catalogue storage is implemented over an external MongoDB database, which due to its schema-less design allows the integration of multiple types of digital objects that may only share their administrative metadata in a single collection that does not need to be modified if a new type of digital object, with previously unencountered properties needs to stored. The personal digital repository shared context is implemented using an embedded graph database with an adapter that allows it to be treated as if it were a triple store. While the use of MongoDB imposes an additional component that must be maintained by the repository owner, the use of an embedded graph database allows it to be distributed with the repository itself, thus slightly reducing the number of independent components of the personal digital repository itself. To supply content for the personal digital repository, four content collection agents and their respective content support modules for the personal digital repository were also developed. Together these form the intertwined system of collection agents and personal repositories that was proposed in the personal digital repository's architecture. Each of the developed content collection agents was geared to collect non traditional content, more specifically the browser history, text messages and contacts from mobile devices, social media posts and chat messages. They follow the template proposed in the architecture where they provide an ancillary service that serves to attract users, while at the same time offering additional services that require a personal digital repository as a support medium in order to be used. When multiple content gathering agents were used in tandem, it was possible to see the relations between content pieces (in the presented example identifying who suggested the visit to a particular web page). Nevertheless the reference implementation came with the downside of not having a finalised user interface. While the interface to administer modules was in place, the lack of a fully fleshed navigation interface hinders any efforts by the repository owners to use it as one stop recall station. Nevertheless it should be noted that the primary objective of this implementation was to serve as a testbed for the underlying data storage and module operations, and in that regard it performed its task adequately.

The proposed architecture is not without its flaws. Determining how to deploy a personal digital repository system remains an open subject. Truly personal repositories should be under the complete control of its owner. This includes complete control over the software stack and the physical hardware upon which it runs. The issue with such a deployment strategy is that not all potential repository users will have the technical inclination or knowledge to maintain the system running over long periods of time. This effect can be compounded by effects of shifting interests that may turn the maintenance of a personal digital repository from an hobby to a chore and can lead to its eventual abandonment. While the ancillary functionalities of the content collection tools may provide a reason to keep its use, it is unwise to think that those functionalities will not appear in other software packages that demand less of a temporal investment. This issue may be mitigated if the repository owners are willing to forego complete control over the repository in exchange for convenience. In that case, a more interesting approach is to offer the personal digital repository as it it were a cloud service. In

this scenario repository owners become clients of a third party entity that takes responsibility for the maintenance of the hardware and software stack. The rise in popularity of shared cloud based content sharing platforms might indicate that users would be willing to trade complete control for ease of use. Yet this trade is not without drawbacks, as it would expose its uses to the same systemic risks that exist in other public platforms (i.e. the content may suddenly become at risk due to the disappearance of the steward company). Furthermore by not controlling everything it becomes even more legitimate for potential repository owners to question if they should trust the proposed system of personal digital repositories. After all, a side effect of attempting to make content gathering the least intrusive possible is that some of that collection may appear to be done using a stealth approach. This is even more true if one takes into account that the chief content collection strategy depends on content collection agents that are essentially doing a bait-and-switch tactic, that lures users with the promise of a service and then presents them with another one that they may not want (and where the content collection takes place). Even if the content collection agents state in their descriptions that content will be transferred to a personal digital repository (that may or may not be under the complete control of the repository owner) there will always those who see in such a dissimulation tactics reason to not trust the system. Despite any technological warranties that may be built in the personal digital repository system itself, or any declarations by a potential company that sponsor the system regard the privacy and ownership o the collected content, the fact is that if an individual chooses not to trust the personal digital repository system, there is little that can be done as in the end trust, though it might be encouraged by external factors, will always be a matter of personal belief. Another possible contention point comes from the suitability of the model chosen to serve as support for the personal digital repository shared context. It can be argued that even without any extensions the CIDOC/CRM model is verbose and overly complex for the task at hand. This is compounded by the fact that the model uses an monotonic approach that allows the accumulation of facts that may contradict each other. Regarding this last trait, it should be noted that personal scenarios will be rife with inconsistencies. These may come from many sources, ranging from the changing interests and view points of the individuals, to the different representations that they construct about themselves in different social contexts (and that spill to social media to be gathered in the personal digital repository) to even some facts that individuals at a given time believed to be true and that may end up revealing as false later. All of these scenarios are capable of generating information that has the potential to contradict the one already present in the personal digital repository shared context, yet this does not necessarily mean that at a the time the information was produced it was already incorrect. The repository owner might have not known better. For instance it may have thought that another individual had a particular family relation with another and later discovered that wasn't the case and corrected its view. Belief in that "fact" might have influence in the production of other content pieces. As long as both the initial information and the corrected one are temporally grounded, which given the CIDOC/CRM emphasis on temporal information is a fair assumption to be made, and there is way to indicate what is the current state that the repository owner believes in, it is preferable to allow these seemingly contradicting pieces of information to accumulate in the shared context than to provide a completely sanitised view. This ends up contributing to the issue of the model's complexity in relation to the personal use case. While it is true that its complexity will have costs, particularly in how queries addressed to the shared context must be formulated, it is required to be able to deal with such nuances in changing data. One can also argue that the roots of the CIDOC/CRM model are on physical objects and that those

parts of the model will hardly be used in a scenario geared towards digital objects (so much that it needed to be extended to better represent them) only increase the complexity without adding anything in return. Yet despite its digital status, some digital content produced by individuals will still be connected to the physical world. Either because an event took place at a given location (such as the sending or receiving of text messages) or because the content directly references physical objects that may also need to be represented, the physical part of the CIDOC/CRM model can eventually be useful. Being able to represent it makes the shared context more flexible, thought at the cost of added complexity. Additionally, it can also be argued that its organisation approach is far removed from how individuals organise their own content, thus creating an impedance mismatch between the repository owner's expectations (if any) and what is represented in the shared context. During the definition of the proposed extension to the CIDOC/CRM mode, the type system was purposely left unused so that in the future repository owners may craft their own organisation schema that can be overlaid on top of the adopted model, thus attempting to minimising the effects of that impedance mismatch. Furthermore, it should be the task of available repository modules to provide additional content visualisations that can also minimise this effect (though it should be noted that none was implemented in the described reference implementation). Finally, in spite of being aimed at individuals, a personal digital repository can not stand alone. Even without a formal community to act as its steward, there will always be multiple individuals with shared interests. The development of new collection agents and their respective counterpart content modules in the repository or of service or visualisation modules will still require personnel with the appropriate skills. The development of such sub-communities will be critical for increasing the types of content with which a system of personal digital repositories can deal.

Overall the proposed approach of an ecosystem composed of multiple content collection agents placed as near to the content sources as possible, working in connection with a personal digital repository represents an alternative to traditional content gathering and selection approaches. Relying on automated agents to gather content in a way that minimises its impact on the users routines might be the key to gather large amounts of content. Content selection is effectively done by the choice of content gathering agents done by the repository owner. This means that there is always the risk of missing on what might eventually be important content due to the content collection tools themselves not being attractive enough to warrant their use. The personal digital repository design focused primary on metadata, leaving the content proper to a secondary position. This means that it does not want to become an authoritative source for it, but instead a gateway that can lead its owner back to possibly lost content. it also highlights its reliance on existing cloud services (to the point where it can be deployed as one) thrusting them to keep content that is not immediately at risk or exploiting them to create multiple copies of collected content. The personal digital repository attempts lead its owner back to potentially forgotten content by creating a shared context upon which the shared relations (be them direct or indirect) between disparate pieces of content can be made explicit, so that tangential starting points, such as persons or events can eventually lead to associated content (web pages suggested by them) or vice versa. The personal digital repository is initially hidden behind its content collection agents. They serve as its initial interface, though one that might be limited by the agent's own scope. With time it is expected that the repository owner's content location efforts start to be directed at the personal digital repository itself, where he can then take advantage of any navigation modules available or of its search capabilities.

For the future, the proposed reference implementation needs to keep evolving. It requires

the development and fielding of additional content visualisation modules that can take advantage of the shared context to allow different kinds of navigations instead of relying on raw content hopping. These can include temporal based navigations (such a time-line from which content and events can be "hanged") or those based on user defined hierarchy (thus forming a personal "related content" mesh) or even social clustering (with content that has some relation to other individuals, such as messages, being presented clustered around them). Other types of content collection agents will also need to be created, being that the most pressing is probably an email content collection agent, for instance in the form of a plugin for an open source email client. A file synchronisation tool that can use the personal digital repository as cloud storage (in the vein of dropbox) should also be developed in order to deal with more traditional file based content. After these enhancements the next step would be a public test deployment of the personal digital repository, offering it as a service for a limited amount of voluntary participants, in order to test medium term (for example an year) engagement and retention levels, as well as a means to estimate the amount of space needed for a year's worth of content, with particular emphasis on the metadatada storage (both the digital object catalogue and shared context). The underlying organisation model would also need to keep evolving, as the development of new content collection agents will eventually means that it will be confronted with unpredicted content types. Initially these can be dealt by extensions provided by repository modules, yet should these prove to be popular it might be wise to merge them with the personal digital repository ontology itself. Furthermore, despite the existing mappings to other ontologies inherited from the CIDOC/CRM model, it stands to reason that some parts of the proposed extension (and of possible extensions to the extension) will benefit from having their own mappings to domain specific ontologies. This should be done as part of the continued development of the personal digital repository as a whole.

"It's said war - war never changes. Men do, through the roads they walk. And this road... has reached its end."

Ulysses (Character)

Fallout New Vegas: Lonesome Road

References

[1] J. Sutton, "Memory," in *The Stanford Encyclopedia Of Philosophy*, winter 2012 ed., E. N. Zalta, Ed., 2012. [Online]. Available: http://plato.stanford.edu/archives/win2012/entries/memory/#ExtMem last accessed on 2015-12-14.

[2] J. Foer, "Remember this: In the archives of the brain our lives linger or disappear." *National Geographic Magazine*, 2007. [Online]. Available: http://ngm.nationalgeographic.com/2007/11/memory/foer-text/1 last accessed on 2015-12-14.

[3] W. Isaacson, *Einstein: His Life and Universe*. Simon & Schuster, 2007, p. 299.

[4] R. E. Bohn and J. E. Short, "How much information? 2009. report on american consumers," Global Information Industry Center. University of California, San Diego, Tech. Rep., 2009. [Online]. Available: http://ddp.nist.gov/refs/HMI_2009_ConsumerReport_Dec9_2009.pdf last accessed on 2015-12-14.

[5] M. Hilbert and P. López, "The World's Technological Capacity to Store, Communicate, and Compute Information," *Science*, vol. 332, no. 6025, pp. 60–65, Apr. 2011. [Online]. Available: http://www.sciencemag.org/content/332/6025/60 last accessed on 2015-12-14.

[6] J. Rosen, "The web means the end of forgetting," *The New York Times Magazine*, 2010. [Online]. Available: http://www.nytimes.com/2010/07/25/magazine/25privacy-t2.html last accessed on 2015-12-14.

[7] P. Thibodeau, "Cerf sees a problem: Today's digital data could be gone tomorrow," *Computerworld.com*, Jun. 2013. [Online]. Available: http://www.computerworld.com/s/article/9239790 last accessed on 2015-12-14.

[8] T. Kuny, "The digital dark ages? Challenges in the preservation of electronic information," *International Preservation News*, pp. 8–13, 1998. [Online]. Available: http://archive.ifla.org/IV/ifla63/63kuny1.pdf last accessed on 2015-12-14.

[9] H. Furness, "Hay festival 2013: Teenagers' mistakes will stay with them forever, warns google chief Eric Schmidt," *The Telegraph*, May 2013. [Online]. Available: http://www.telegraph.co.uk/technology/eric-schmidt/10080596/Hay-Festival-2013-Teenagers-mistakes-will-stay-with-them-forever-warns-Google-chief-Eric-Schmidt.html last accessed on 2015-12-14.

[10] European Commission, "Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation)," Jan. 2012. [Online]. Available: http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf last accessed on 2015-12-14.

[11] The Nielsen Company, "State of the media: Social media report 2012," 2012. [Online]. Available: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/The-Social-Media-Report-2012.pdf last accessed on 2015-12-14.

[12] Experian, "Experian marketing services reveals 27 percent of time spent online is on social networking," Apr. 2013. [Online]. Available: https://www.experianplc.com/media/news/2013/experian-marketing-services-reveals-27-percent-of-time-spent-online-is-on-social-networking/ last accessed on 2015-12-14.

[13] Consultative Committee for Space Data Systems, "Reference model for an open archival information system (OAIS). Magenta Book," Tech. Rep., Jun. 2012. [Online]. Available: http://public.ccsds.org/publications/archive/650x0m2.pdf last accessed on 2015-12-14.

[14] L. Candela, D. Castelli, N. Ferro, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt, *The DELOS Digital Library Reference model. Foundations for digital Libraries (Version 0.98)*. DELOS: a Network of Excellence on Digital Libraries, 2008. [Online]. Available: http://www.researchgate.net/publication/200462045_The_DELOS_Digital_Library_Reference_Model_-_Foundations_for_Digital_Libraries last accessed on 2015-12-14.

[15] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp, "Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 270–312, Apr. 2004. [Online]. Available: http://doi.acm.org/10.1145/984321.984325 last accessed on 2015-12-14.

[16] U. Murthy, D. Gorton, R. Torres, M. Gonçalves, E. Fox, and L. Delcambre, "Extending the 5s digital library (dl) framework: From a minimal dl towards a dl reference model," in *In Proceedings of the 1st Workshop on Digital Library Foundations, ACM IEEE Joint Conference on Digital Libraries Vancouver, British*, 2007, pp. 25–30.

[17] Y. Ma, E. A. Fox, and M. A. Gonçalves, "Personal digital library: pim through a 5s perspective," in *Proceedings of the ACM first Ph.D. workshop in CIKM*, ser. PIKM '07. New York, NY, USA: ACM, 2007, pp. 117–124. [Online]. Available: http://doi.acm.org/10.1145/1316874.1316893 last accessed on 2015-12-14.

[18] L. L. Barker and G. Wiseman, "A model of intrapersonal communication," *Journal of Communication*, vol. 16, no. 3, pp. 172–179, 1966. [Online]. Available: http://dx.doi.org/10.1111/j.1460-2466.1966.tb00031.x last accessed on 2015-12-14.

[19] C. V. Roberts, K. W. Watson, and L. L. Barker, *Intrapersonal communication processes: original essays*. SPECTRA, 1989. [Online]. Available: http://books.google.pt/books?id=pzUNAQAAMAAJ last accessed on 2015-12-14.

[20] R. M. Gagné and K. Medsker, *The conditions of learning: Training applications*. Harcourt Brace College Pub., 1996. [Online]. Available: http://books.google.pt/books?id=0FFaAAAAYAAJ last accessed on 2015-12-14.

[21] IFLA Study Group on the Functional Requirements for Bibliographic Records and International Federation of Library Associations and Institutions. Section on Cataloguing. Standing Committee, *Functional requirements for bibliographic records: final report*, ser. UBCIM publications. K.G. Saur, 1998. [Online]. Available: http://www.ifla.org/publications/functional-requirements-for-bibliographic-records last accessed on 2015-12-14.

[22] Standing Committee Of The IFLA Cataloguing Section, *ISBD: International Standard Bibliographic Description: Consolidated Edition*, ser. IFLA Series on Bibliographic Control. de Gruyter, Jun. 2011. [Online]. Available: http://www.ifla.org/publications/isbd-international-standard-bibliographic-description-consolidated-edition last accessed on 2015-12-14.

[23] IFLA Working Group on Functional Requirements and Numbering of Authority Records (FRANAR), *Functional Requirements for Authority Data: A Conceptual Model*, ser. Ifla Series on Bibliographic Control, G. E. Patton, Ed. K. G. Saur, 2009. [Online]. Available: http://www.library.illinois.edu/export/cam/rda/files/FRAD.pdf last accessed on 2015-12-14.

[24] IFLA Working Group on the Functional Requirements for Subject Authority Records (FRSAR), *Functional Requirements for Subject Authority Data (FRSAD): A Conceptual Model*, ser. IFLA Series on Bibliographic Control, M. Zeng, M. Žumer, and A. Salaba, Eds. De Gruyter, 2011. [Online]. Available: http://www.ifla.org/files/assets/classification-and-indexing/functional-requirements-for-subject-authority-data/frsad-final-report.pdf last accessed on 2015-12-14.

[25] C. Lagoze and J. Hunter, "The abc ontology and model," *International Conference on Dublin Core and Metadata Applications*, 2001. [Online]. Available: http://dcpapers.dublincore.org/pubs/article/view/655 last accessed on 2015-12-14.

[26] T. Baker, "A grammar of dublin core," *D-Lib Magazine*, vol. 6, no. 10, Oct. 2000. [Online]. Available: http://www.dlib.org/dlib/october00/baker/10baker.html last accessed on 2015-12-14.

[27] *Europeana Data Model Primer*, Europeana, Oct. 2011. [Online]. Available: http://pro.europeana.eu/documents/900548/770bdb58-c60e-4beb-a687-874639312ba5 last accessed on 2015-12-14.

[28] B. Haslhofer and A. Isaac, "data.europeana.eu - the europeana linked open data pilot," in *DC-2011, The Hague*, Aug. 2011. [Online]. Available: http://dcevents.dublincore.org/index.php/IntConf/dc-2011/paper/view/55 last accessed on 2015-12-14.

[29] *ORE Specification - Abstract Data Model*, Open Archives Initiative, 2008. [Online]. Available: http://www.openarchives.org/ore/datamodel last accessed on 2015-12-14.

[30] *Definition of the CIDOC Conceptual Reference Model*, International Council Of Museums, Nov. 2012. [Online]. Available: http://www.cidoc-crm.org/docs/cidoc_crm_version_5.1.pdf last accessed on 2015-12-14.

[31] S. Stead. (2008, Nov.) Bilingual multimedia tutorial for iso 21127. CIDOC. [Online]. Available: http://www.cidoc-crm.org/cidoc_tutorial/index.html last accessed on 2015-12-14.

[32] M. Doerr, "Mapping of the Dublin Core metadata element set to the CIDOC CRM," Tech. Rep., jun 2000, Technical Report FORTH-ICS/TR-274. [Online]. Available: http://www.cidoc-crm.org/docs/dc_to_crm_mapping.pdf last accessed on 2015-12-14.

[33] C. Kakali, I. Lourdi, T. Stasinopoulou, L. Bountouri, C. Papatheodorou, M. Doerr, and M. Gergatsoulis, "Integrating dublin core metadata for cultural heritage collections using ontologies," *International Conference on Dublin Core and Metadata Applications*, 2007. [Online]. Available: http://dcpapers.dublincore.org/pubs/article/view/871 last accessed on 2015-12-14.

[34] M. Theodoridou, Y. Tzitzikas, M. Doerr, Y. Marketakis, and V. Melessanakis, "Modeling and querying provenance by extending cidoc crm," *Distributed and Parallel Databases*, vol. 27, pp. 169–210, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10619-009-7059-2 last accessed on 2015-12-14.

[35] R. Tansley and S. Harnad, "Eprints.org Software for Creating Institutional and Individual Open Archives," *D-Lib Magazine*, vol. 6, no. 10, 2000. [Online]. Available: http://www.dlib.org/dlib/october00/10inbrief.html#HARNAD last accessed on 2015-12-14.

[36] M. Smith, "Dspace: An institutional repository from the mit libraries and hewlett packard laboratories," in *Research and Advanced Technology for Digital Libraries*, ser. Lecture Notes in Computer Science, M. Agosti and C. Thanos, Eds. Springer Berlin Heidelberg, 2002, vol. 2458, pp. 543–549. [Online]. Available: http://dx.doi.org/10.1007/3-540-45747-X_40 last accessed on 2015-12-14.

[37] C. Lagoze, S. Payette, E. Shin, and C. Wilper, "Fedora: an architecture for complex objects and their relationships," *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 124–138, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1007/s00799-005-0130-3 last accessed on 2015-12-14.

[38] S. Lewis. (2014) Repository66.org repository maps. [Online]. Available: http://maps.repository66.org/ last accessed on 2015-12-14.

[39] L. L. Peterson, "E-ternally yours: The case for the development of a reliable repository for the preservation of personal digital objects," Undergraduate Honors Thesis, College of Engineering and Computer Science, University of Central Florida, Orlando, Florida, 2010. [Online]. Available: http://explorer.cyberstreet.com/CET4970H-Peterson-Thesis.pdf last accessed on 2015-12-14.

[40] M. Leggott, "Islandora: a drupal/fedora repository system," 4th International Conference on Open Repositories, Atlanta, Georgia, USA, May 2009. [Online]. Available: http://hdl.handle.net/1853/28495 last accessed on 2015-12-14.

[41] D. Gourley and P. B. Viterbo, "A sustainable repository infrastructure for digital humanities: The dho experience." in *EuroMed*, ser. Lecture Notes in Computer Science, M. Ioannides, D. W. Fellner, A. Georgopoulos, and D. G. Hadjimitsis, Eds., vol. 6436. Springer, 2010, pp. 473–481. [Online]. Available: http://dblp.uni-trier.de/db/conf/euromed/euromed2010.html#GourleyV10 last accessed on 2015-12-14.

[42] S. Prater. (2014, Nov.) Fedora digital object model. DuraSpace. [Online]. Available: https://wiki.duraspace.org/display/FEDORA38/Fedora+Digital+Object+Model last accessed on 2015-12-14.

[43] R. Kahn and R. Wilensky, "A framework for distributed digital object services," *Int. J. Digit. Libr.*, vol. 6, no. 2, pp. 115–123, 2006, last accessed on 2015-12-14.

[44] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," W3C Recommendation, Jan. 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/ last accessed on 2015-12-14.

[45] S. Thomas, "A practical approach to the preservation of personal digital archives," Paradigm project, Tech. Rep., Mar. 2007. [Online]. Available: http://www.paradigm.ac.uk/workbook/index.html last accessed on 2015-12-14.

[46] Paradigm project, *Workbook on Digital Private Papers*, 2005-7. [Online]. Available: http://www.paradigm.ac.uk/workbook/index.html last accessed on 2015-12-14.

[47] S. R. Kruk, S. Decker, and L. Zieborak, "Jeromedl - adding semantic web technologies to digital libraries," 2005, pp. 716–725. [Online]. Available: http://core.ac.uk/download/files/300/10851867.pdf last accessed on 2015-12-14.

[48] S. R. Kruk, M. Synak, and K. Zimmermann, "Marcont: integration ontology for bibliographic description formats," in *DCMI '05: Proceedings of the 2005 international conference on Dublin Core and metadata applications.* Dublin Core Metadata Initiative, 2005, pp. 1–5.

[49] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.99," Namespace document, Jan. 2014. [Online]. Available: http://xmlns.com/foaf/spec/ last accessed on 2015-12-14.

[50] F. Alvarez-Cavazos, D. A. Garza-Salazar, and J. C. Lavariega-Jarquin, "Pdlib: personal digital libraries with universal access," in *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries.* New York, NY, USA: ACM, 2005, pp. 365–365. [Online]. Available: http://doi.acm.org/10.1145/1065385.1065468 last accessed on 2015-12-14.

[51] F. Alvarez-Cavazos, R. Garcia-Sanchez, D. Garza-Salazar, J. C. Lavariega, L. G. Gomez, and M. Sordia, "Universal access architecture for digital libraries," in *CASCON '05: Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research.* IBM Press, 2005, pp. 12–28.

[52] *The Open Archives Initiative Protocol for Metadata Harvesting*, Open Archives Initiative, Jun. 2002. [Online]. Available: http://www.openarchives.org/OAI/openarchivesprotocol.html last accessed on 2015-12-14.

[53] W. C. Janssen and K. Popat, "Uplib: a universal personal digital library system," in *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*. New York, NY, USA: ACM, 2003, pp. 234–242. [Online]. Available: http://doi.acm.org/10.1145/958220.958262 last accessed on 2015-12-14.

[54] W. C. Janssen, "The uplib personal digital library system," in *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*. New York, NY, USA: ACM, 2005, pp. 410–410. [Online]. Available: http://doi.acm.org/10.1145/1065385.1065514 last accessed on 2015-12-14.

[55] D. Bainbridge, B. J. Novak, and S. J. Cunningham, "A user-centered design of a personal digital library for music exploration," in *JCDL '10: Proceedings of the 10th annual joint conference on Digital libraries*. New York, NY, USA: ACM, 2010, pp. 149–158. [Online]. Available: http://doi.acm.org/10.1145/1816123.1816145 last accessed on 2015-12-14.

[56] D. Bainbridge, B. Novak, and S. J. Cunningham, "A spatial hypertext-based, personal digital library for capturing and organizing musical moments," *International Journal on Digital Libraries*, vol. 12, no. 2-3, pp. 89–103, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00799-012-0090-3 last accessed on 2015-12-14.

[57] H. Figueirêdo, Y. Lacerda, A. Paiva, M. Casanova, and C. Souza Baptista, "Photogeo: a photo digital library with spatial-temporal support and self-annotation," *Multimedia Tools and Applications*, vol. 59, no. 1, pp. 279–305, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11042-011-0745-x last accessed on 2015-12-14.

[58] S. Strodl, F. Motlik, K. Stadler, and A. Rauber, "Personal & soho archiving," in *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*. New York, NY, USA: ACM, 2008, pp. 115–123. [Online]. Available: http://doi.acm.org/10.1145/1378889.1378910 last accessed on 2015-12-14.

[59] S. Strodl, P. Petrov, M. Greifeneder, and A. Rauber, "Automating logical preservation for small institutions with hoppla," in *ECDL*, 2010, pp. 124–135. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15464-5_14 last accessed on 2015-12-14.

[60] M. Vandor. (2012, Dec.) Your Twitter archive. Twitter. [Online]. Available: https://blog.twitter.com/2012/your-twitter-archive last accessed on 2015-12-14.

[61] (2014) Accessing your facebook data. Facebook. [Online]. Available: https://www.facebook.com/help/405183566203254 last accessed on 2015-12-14.

[62] (2011, Jun.) The data liberation front delivers google takeout. Google. [Online]. Available: http://dataliberation.blogspot.pt/2011/06/data-liberation-front-delivers-google.html last accessed on 2015-12-14.

[63] L. C. Aun. (2013, Nov.) Life. [Online]. Available: https://github.com/cheeaun/life last accessed on 2015-12-14.

[64] A. Baddeley, "Working Memory: Theories, Models, and Controversies," in *ANNUAL REVIEW OF PSYCHOLOGY, VOL 63*, ser. Annual Review of Psychology, Fiske, ST and Schacter, DL and Taylor, SE, Ed. 4139 EL CAMINO WAY, PO BOX 10139, PALO ALTO, CA 94303-0897 USA: ANNUAL REVIEWS, 2012, vol. 63, pp. 1–29. [Online]. Available: http://www.annualreviews.org/doi/abs/10.1146/annurev-psych-120710-100422 last accessed on 2015-12-14.

[65] R. Conrad, "Acoustic confusions in immediate memory," *British Journal of Psychology*, vol. 55, no. 1, pp. 75–84, 1964. [Online]. Available: http://dx.doi.org/10.1111/j.2044-8295.1964.tb00899.x last accessed on 2015-12-14.

[66] A. D. Baddeley, "The influence of acoustic and semantic similarity on long-term memory for word sequences," *Quarterly Journal of Experimental Psychology*, vol. 18, no. 4, pp. 302–309, 1966. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/14640746608400047 last accessed on 2015-12-14.

[67] D. Gonçalves and J. A. Jorge, "In search of personal information: Narrative-based interfaces," in *Proceedings of the 13th International Conference on Intelligent User Interfaces*, ser. IUI '08. New York, NY, USA: ACM, 2008, pp. 179–188. [Online]. Available: http://doi.acm.org/10.1145/1378773.1378797 last accessed on 2015-12-14.

[68] J. Gemmell, G. Bell, and R. Lueder, "Mylifebits: A personal database for everything," *Commun. ACM*, vol. 49, no. 1, pp. 88–95, Jan. 2006. [Online]. Available: http://doi.acm.org/10.1145/1107458.1107460 last accessed on 2015-12-14.

[69] V. Bush, "As We May Think," *Atlantic Monthly*, vol. 176, no. 1, pp. 641–649, Mar. 1945. [Online]. Available: http://www.theatlantic.com/doc/194507/bush last accessed on 2015-12-14.

[70] G. Bell and J. Gemmell, "A digital life," *Scientific American Magazine*, vol. 296, no. 3, pp. 58–65, 2007.

[71] G. Greenwald and E. MacAskill. (2013, Jun.) NSA prism program taps in to user data of Apple, Google and others. [Online]. Available: http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data last accessed on 2015-12-14.

[72] S. Mann, J. Nolan, and B. Wellman, "Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments." *Surveillance & Society*, vol. 1, no. 3, pp. 331–355, 2002.

[73] G. Bell, J. Gemmell, and B. Gates, *Your Life, Uploaded: The Digital Way to Better Memory, Health, and Productivity.* Penguin Group US, 2010.

[74] S. Whittaker and C. Sidner, "Email overload: exploring personal information management of email," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '96. New York, NY, USA: ACM, 1996, pp. 276–283. [Online]. Available: http://doi.acm.org/10.1145/238386.238530 last accessed on 2015-12-14.

[75] W. Jones, S. Dumais, and H. Bruce, "Once found, what then? A study of "keeping" behaviors in the personal use of Web information," *Proceedings of the American Society*

for Information Science and Technology, vol. 39, no. 1, pp. 391–402, 2002. [Online]. Available: http://dx.doi.org/10.1002/meet.1450390143 last accessed on 2015-12-14.

[76] S. Whittaker, V. Bellotti, and J. Gwizdka, "Email in personal information management," *Commun. ACM*, vol. 49, no. 1, pp. 68–73, Jan. 2006. [Online]. Available: http://doi.acm.org/10.1145/1107458.1107494 last accessed on 2015-12-14.

[77] C. Marshall, F. Mccown, and M. Nelson, "Evaluating Personal Archiving Strategies for Internet-based Information," in *Proceedings of Archiving 2007*, 2007, pp. 151–156. [Online]. Available: http://arxiv.org/pdf/0704.3647 last accessed on 2015-12-14.

[78] C. Marshall, "Retinking Personal Digital Archiving, Part 1: Four Challenges from the Field," *D-Lib Magazine*, vol. 14, no. 3, 2008. [Online]. Available: http://www.dlib.org/dlib/march08/marshall/03marshall-pt1.html last accessed on 2015-12-14.

[79] "Forecast: Devices by operating system and user type, worldwide, 2010-2017 1q13 update," Gartner, Tech. Rep., Apr. 2013. [Online]. Available: http://www.gartner.com/resId=2396815 last accessed on 2013-07-14.

[80] (2013, Apr.) Gartner says worldwide pc, tablet and mobile phone combined shipments to reach 2.4 billion units in 2013. Gartner. [Online]. Available: http://www.gartner.com/newsroom/id/2408515 last accessed on 2015-12-14.

[81] S. Garfinkel and D. Cox, "Finding and archiving the internet footprint," 2009. [Online]. Available: http://simson.net/clips/academic/2009.BL.InternetFootprint.pdf last accessed on 2015-12-14.

[82] D. Sinn, S. Syn, and K. Sung-Min, "Personal records on the web: Who's in charge of archiving, Hotmail or archivists?" *Library & Information Science Research*, vol. 33, no. 4, pp. 320–330, 2011. [Online]. Available: http://sciencedirect.com/science/article/pii/S0740818811000624 last accessed on 2015-12-14.

[83] G. Greenwald. (2013, Aug.) Glenn greenwald: detaining my partner was a failed attempt at intimidation. The Guardian. [Online]. Available: http://www.theguardian.com/commentisfree/2013/aug/18/david-miranda-detained-uk-nsa last accessed on 2015-12-14.

[84] W. Moncur, J. Bikker, E. Kasket, and J. Troyer, "From death to final disposition: Roles of technology in the post-mortem interval," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 531–540. [Online]. Available: http://doi.acm.org/10.1145/2207676.2207750 last accessed on 2015-12-14.

[85] E. Kasket, "Being-towards-death in the digital age." *Existential Analysis: Journal of the Society for Existential Analysis*, vol. 23, no. 2, pp. 249 – 261, 2012. [Online]. Available: http://connection.ebscohost.com/c/articles/82404657/being-towards-death-digital-age last accessed on 2015-12-14.

[86] L. Maffeo. (2013, Nov.) Ghost in the Cloud: Dealing with data after death. The Next Web. [Online]. Available: http://thenextweb.com/socialmedia/2013/11/16/ghost-cloud-dealing-data-death last accessed on 2015-12-14.

[87] (2013) Deadsocial. DeadSocial. [Online]. Available: http://www.deadsocial.org/ last accessed on 2015-12-14.

[88] (2012) Everplan. Everplans. [Online]. Available: https://www.everplans.com/ last accessed on 2015-12-14.

[89] (2010) Virtual eternity. Virtual Eternity. [Online]. Available: http://www.virtualeternity.com last accessed on 2013-07-22.

[90] (2013) Liveson. [Online]. Available: http://liveson.org last accessed on 2015-12-14.

[91] (2013) Eter9. [Online]. Available: http://eter9.com last accessed on 2015-12-14.

[92] (2013) About inactive account manager. Google. [Online]. Available: https://support.google.com/accounts/answer/3036546?hl=en&ref_topic=3075532 last accessed on 2015-12-14.

[93] K. Hill. (2015, Apr.) This start-up promised 10,000 people eternal digital life—then it died. [Online]. Available: http://fusion.net/story/116999/this-start-up-promised-10000-people-eternal-digital-life-then-it-died/ last accessed on 2015-12-14.

[94] D. Lee. (2013, Sep.) Facebook apologies for dating ad showing rehtaeh parsons. BBC. [Online]. Available: http://www.bbc.com/news/technology-24141835 last accessed on 2015-12-14.

[95] Aristotle, T. Sinclair, and T. Saunders, *The Politics*, ser. Penguin classics, T. Sinclair and T. Saunders, Eds. Penguin Books Limited, 1981.

[96] K. Marx, *Selected Writings*, ser. Classics Series, L. Simon, Ed. Hackett, 1994.

[97] S. N. Young, "The neurobiology of human social behaviour: an important but neglected topic," *Journal of Psychiatry and Neuroscience*, vol. 33, no. 5, pp. 391–392, Sep. 2008. [Online]. Available: http://jpn.ca/vol33-issue5/ last accessed on 2015-12-14.

[98] D. Matsumoto, "Culture, context and behavior," *Journal of Personality*, vol. 75, no. 6, pp. 1285–1320, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1467-6494.2007.00476.x last accessed on 2015-12-14.

[99] K. Bollacker, "Avoiding a digital dark age," *American Scientist*, vol. 98, no. 2, Apr. 2010. [Online]. Available: http://www.americanscientist.org/issues/num2/2010/3/avoiding-a-digital-dark-age/1 last accessed on 2015-12-14.

[100] V. Reich and D. S. H. Rosenthal, "LOCKSS: A permanent web publishing and access system," *DLib Magazine*, vol. 7, no. 6, 2001. [Online]. Available: http://www.dlib.org/dlib/june01/reich/06reich.html last accessed on 2015-12-14.

[101] W. C. Dougherty, "Can digital resources truly be preserved?" *The Journal of Academic Librarianship*, vol. 36, no. 5, pp. 445–448, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0099133310001795 last accessed on 2015-12-14.

[102] (2013, Nov.) New digital preservation alliance links the uk, ireland and the netherlands. Digital Preservation Coalition. [Online]. Available: http://www.dpconline.org/newsroom/latest-news/1111-new-digital-preservation-alliance last accessed on 2015-12-14.

[103] S. Needham and T. Spence, "Refuse and the formation of middens," *Antiquity*, vol. 71, no. 271, pp. 77–90, 1997.

[104] S. Zhao, S. Grasmuck, and J. Martin, "Identity construction on facebook: Digital empowerment in anchored relationships," *Computers in Human Behavior*, vol. 24, no. 5, pp. 1816 – 1836, 2008, including the Special Issue: Internet Empowerment. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563208000204 last accessed on 2015-12-14.

[105] V. Mayer-Schönberger, *Delete: The Virtue of Forgetting in the Digital Age*. Princeton University Press, 2011.

[106] C. Kotfila, "This message will self-destruct: The growing role of obscurity and self-destructing data in digital communication," *Bulletin of the American Society for Information Science and Technology*, vol. 40, no. 2, pp. 12–16, 2014. [Online]. Available: http://dx.doi.org/10.1002/bult.2014.1720400206 last accessed on 2015-12-14.

[107] J. B. Bayer, N. B. Ellison, S. Y. Schoenebeck, and E. B. Falk, "Sharing the small moments: ephemeral social interaction on snapchat," *Information, Communication & Society*, vol. 0, no. 0, pp. 1–22, 0. [Online]. Available: http://yardi.people.si.umich.edu/pubs/Schoenebeck_Snapchat15.pdf last accessed on 2015-12-14.

[108] J. Barateiro, G. Antunes, F. Freitas, and J. L. Borbinha, "Designing digital preservation solutions: A risk management-based approach." *IJDC*, vol. 5, no. 1, pp. 4–17, 2010. [Online]. Available: http://www.ijdc.net/index.php/ijdc/article/view/143 last accessed on 2015-12-14.

[109] S. Vermaaten, B. Lavoie, and P. Caplan, "Identifying threats to successful digital preservation: the spot model for risk assessment." *D-Lib Magazine*, vol. 18, no. 9/10, 2012. [Online]. Available: http://www.dlib.org/dlib/september12/vermaaten/09vermaaten.html last accessed on 2015-12-14.

[110] "Understanding metadata," *National Information Standards*, vol. 20, 2004. [Online]. Available: http://www.niso.org/publications/press/UnderstandingMetadata.pdf last accessed on 2015-12-14.

[111] B. Leiba, "Oauth web authorization protocol." *IEEE Internet Computing*, vol. 16, no. 1, pp. 74–77, 2012.

[112] J. Richer, M. B. Jones, J. Bradley, M. Machulak, and P. Hunt. (2015, Jul.) Oauth 2.0 dynamic client registration protocol. Internet Engineering Task Force. [Online]. Available: https://tools.ietf.org/html/rfc7591 last accessed on 2015-12-14.

[113] M. Bleigh. (2010, Apr.) Rest isn't what you think it is, and that's ok. [Online]. Available: https://www.mobomo.com/2010/04/rest-isnt-what-you-think-it-is/ last accessed on 2015-12-14.

[114] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011, http://d3js.org/. [Online]. Available: http://vis.stanford.edu/papers/d3 last accessed on 2015-12-14.

[115] A. Brown, "The pronom puid scheme: A scheme of persistent unique identifiers for representation information," *London: The National Archives*, 2006. [Online]. Available: http://www.nationalarchives.gov.uk/PRONOM/ last accessed on 2015-12-14.

[116] J. Klensin, T. Hansen, and N. Freed. (2013, Jan.) Media type specifications and registration procedures. Internet Engineering Task Force. [Online]. Available: https://tools.ietf.org/html/rfc6838 last accessed on 2015-12-14.

[117] V. Callison-Burch, J. Probst, and M. Govea. (2015, Feb.) Adding a legacy contact. Facebook. [Online]. Available: http://newsroom.fb.com/news/2015/02/adding-a-legacy-contact/ last accessed on 2015-12-14.

[118] (2015) icloud terms and conditions. Apple. [Online]. Available: http://www.apple.com/legal/internet-services/icloud/en/terms.html last accessed on 2015-12-14.

[119] (2015) Cirrus legacy. Cirrus Legacy. [Online]. Available: http://www.cirruslegacy.com/ last accessed on 2015-12-14.

[120] C. Emil. (2014, Aug.) Groups and relations between persons. CIDOC/CRM. [Online]. Available: http://139.91.183.82:8888/drupal/Issue/groups-and-relations-between-persons last accessed on 2015-12-14.

[121] (2005) The grails framework. The Grails Project. [Online]. Available: https://grails.org/ last accessed on 2015-12-14.

[122] (2003) Groovy. The Groovy Project. [Online]. Available: http://www.groovy-lang.org/ last accessed on 2015-12-14.

[123] (1995) Java. Oracle. [Online]. Available: https://www.oracle.com/java/index.html last accessed on 2015-12-14.

[124] (1999) Apache tomcat. Apache Software Foundation. [Online]. Available: http://tomcat.apache.org/ last accessed on 2015-12-14.

[125] (2009) Mongodb. MongoDB Inc. [Online]. Available: https://www.mongodb.org/ last accessed on 2015-12-14.

[126] (2007) Neo4j. Neo Technology. [Online]. Available: http://neo4j.com/ last accessed on 2015-12-14.

[127] (2010) Orientdb. Orient Technologies LTD. [Online]. Available: http://orientdb.com/orientdb/ last accessed on 2015-12-14.

[128] (2015, Oct.) Blueprints. Thinkerpop. [Online]. Available: https://github.com/tinkerpop/blueprints/wiki last accessed on 2015-12-14.

[129] Graphviz - graph visualization software. [Online]. Available: http://www.graphviz.org/ last accessed on 2015-12-14.

[130] J. Sousa, M. Pereira, and J. A. Martins, "Improving browser history using semantic information," in *ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, Volume 2, Wroclaw, Poland, 28 June - 1 July, 2012*, L. A. Maciaszek, A. Cuzzocrea, and J. Cordeiro, Eds. SciTePress, 2012, pp. 305–311.

[131] R. Soares, M. Pereira, and J. Martins, "Recolha, preservação e contextualização de objectos digitais para dispositivos móveis com android," *Iberian Journal of Information Systems and Technologies*, vol. 0, no. 9, 2012. [Online]. Available: http://ojs.academypublisher.com/index.php/risti/article/view/risti097589 last accessed on 2015-12-14.

[132] T. A. d. C. Azevedo, "Repositório de dados pessoais das redes sociais," Master's thesis, Universidade de Aveiro, 2012. [Online]. Available: http://hdl.handle.net/10773/10942 last accessed on 2015-12-14.

[133] Adium. Adium team. [Online]. Available: http://adium.im/ last accessed on 2015-12-14.

[134] iTunes. Apple. [Online]. Available: https://www.apple.com/itunes/ last accessed on 2015-12-14.

[135] last.fm. CBS Interactive. [Online]. Available: http://www.last.fm/ last accessed on 2015-12-14.

Estes anexos só estão disponíveis para consulta através do CD-ROM.
Queira por favor dirigir-se ao balcão de atendimento da Biblioteca.

Serviços de Biblioteca, Informação Documental e Museologia
Universidade de Aveiro