Evaluation

Complying to the FAIR principles

Earlier, the FAIR principles (Findability, Accessibility, Interoperability and Reusability) were introduced as drawn up by Wilkinson *et al.* in 2016. In the preceding chapter, a complete data model with attributes, flowing from domain-specific requirements, is proposed. Table (?) shows how the data model abstractly complies to specific FAIR principles.

FAIR principle	Data model characteristics (Specific FAIR principle)
Findability	• Publications and documents use a unique identifier (F1, F3) • Publications and documents are enriched with relevant (sometimes required) attributes, such as a date and a source url (F2) • The data are registered in a resource (JSON, or XML) that empowers and enables searchability (F4)
Accessibility	• Publications and documents use a unique identifier that is retrievable through the JSON communications protocol (A1) • Even when the data are no longer available online, the data file (JSON) persists (A2)
Interoperability	• JSON is a formal, shared and broadly applicable language for knowledge representation (I1) • By standardizing the attributes, everybody uses a vocabulary following FAIR principles (I2)
Reusability	• Publications and documents are enriched with relevant attributes (R1) • Attributes are chosen from a broad selection of perspectives, creating a domain-relevant vocabulary (R1.3)

Publication level

To demonstrate the compliance to the FAIR principles on the publication level, suppose a user has the specific information need to find all publications that are not answered in the legal timeframe. The Woo has set the maximum number of days to respond to 42 days. This information need was the central focus of the 'Ondraaglijk Traag' (Dutch for 'unbearable slow') report published by Open State Foundation in 2022. The information need can be formalized into the following query:

Q = object = 'Publication' AND (decisionDate – fileDate) <= 42

It denotes the need for two pieces of information, a decision date and a file date. The dates need to be in some sort of date format to enable making calculations on the dates. By investigating historical publications done by the Dutch government, the conclusion can be drawn that it is practically impossible to extract the decision and file date from a bunch of publications. On the publication page, a date is provided. However, there exists no label to indicate the meaning of this date. The best guess is that this is a publication date, which is not necessarily the same as a decision date. Decision documents always contain the wanted date information, but they are not easily extracted. Named Entity Recognition to classify pieces of text as dates sounds like a solution, but as publications contain multiple kinds of dates, it can be very hard to classify one date as 'decision date' and the other as 'file date'. Also, the dates probably needs preprocessing from a string to a workable date. To bring up the FAIR principles again, specific attributes like the requested dates are not easily accessible, as they are hidden inside a document with a lot of textual data. The lack of metadata makes these attributes unfindable. Reusing the publications for research or monitoring purposes is a time consuming task, which must be done by hand, just as the 'Ondraaglijk Traag' report, to ensure valid results. The solution to fulfilling the information need is the addition of supporting data to the publications. This would be an uncomplicated task, as most of the data are already supplied somewhere in the responses, just not findable and readable for a computer. In addition, validating the publications would ensure that the dates are in the right format for further processing.

Document level

A same kind of example can be thought of for documents. Let us assume that a journalist has the information need to quickly find all email documents about the Dutch ‘Mondkapjesdeal’ (face mask deal). That journalist can formulate his information need as follows:

Q = object = ‘Document’ AND annexType = ‘Email’ AND title CONTAINS ‘mondkapjesdeal’

To investigate the existence of document metadata like the annex type and title, 2703 decisions on information requests will be analyzed originating from the Dutch government’s website. Inventory lists often contain useful information on documents. To identify inventory lists for these publications, documents are classified based on their file name. Files containing the word ‘inventaris’ (inventory) are classified as inventory lists. For only 436 publications there is an automatically identified inventory list available. For other publications, either the file name has no hint of an inventory list, or the inventory list is appended to the decision document, or there is simply no inventory list provided. As all the inventory lists are of the pdf format, the next challenge is identifying tables in the pdf documents. The Python module ‘pdfplumber’ supports table extraction from pdf documents and is used to identify tables. In 90 of the inventory lists, there was no table found at all. This can be caused by, for example, a pdf document not being readable, like searching with control-f is not possible. For the resulting inventory lists with a table, the first row of the table is searched in which at least half of the columns is filled with something, skipping empty rows and title rows which are often present. In 245 of the resulting inventory lists, evidence of the existence of a title is found. At least 8 different ways of denoting a title are found. This was found by investigating the column headers and their occurrences by hand, classifying the headers manually. Only 71 publications showed clear evidence of a ‘type’ attribute, with also 8 ways of indicating a type. To fulfil the information need, already some time consuming tasks are done, leaving us with only 71 of in total 2703 publications to search for emails regarding the face mask deal. The lack of properly providing metadata and the amount of work needed to find them makes the information unfindable and inaccessible. The ambiguous column names illustrate that there is no sign of interoperability. The solution is to enrich the documents with sufficient metadata. By storing the metadata in a shared and broadly accepted language for knowledge representation like JSON, the data are accessed much more easier than in a poorly legible pdf format.

Supporting tables – not yet decided on which ones to use

Total publications	2703
Inventory list identified automatically	436
Table extractable	346
Evidence of a ‘title’	245
Evidence of an ‘annexType’	71

Attribute	Number of times found	Number of options	Options
identifier	260	7	"nr", "nummer", "volgnummer", "docnr", "documentnr", "id", "documentnummer"
title	245	8	"document", "documentnaam", "titeldocument", "titeldoc", "onderwerp", "naamdocument", "titel", "naam"
valuation	233	4	"beoordeling", "beroordeling", "oordeel", "beoordelingwob"
groundsOfRefusal	236	13	"weigeringsgrond", "artikelwob", "wob", "beslissingconform", "wobgrond", "uitzonderingsgrond", "artikel", "wobartikel", "weigeringsgrondwob", "weigeringsgronden", "lakgrond", "relevantewobgronden", "grond"
date	178	2	"datum", "datumdocument"
annexType	71	8	"soort", "soortdocument", "type", "categorie", "documenttype", "typedocument", "soortstuk", "documentsoort"
originator	185	3	"afzender", "afzenders", "van"
recipient	180	4	"ontvanger", "ontvangers", "naar", "aan"