

Inhoud

The schema	2
Requirements.....	2
Technical requirements	2
End-user requirements	2
A semi-structured approach	3
JSON vs. XML.....	5
Attribute selection	5
Metadata standard (Dublin Core).....	5
Journalistic standard (Follow the Money)	6
Earlier work (Open State, VNG, North-Holland)	6
Analysis of current publicly available publications	6
Extractable metadata.....	7
Constraints	Fout! Bladwijzer niet gedefinieerd.
Complying to the FAIR principles	8

The schema

The goal of this research is to represent publications and documents related to information requests or active disclosures of information. This research part tries to fit the publications and documents into a data model with associated attributes. This is done by firstly listing requirements that the data model needs to comply to. Then, a data model with attributes fitting these requirements is introduced.

Wilkinson *et al.* (2016) published guiding principles for data management and stewardship. The FAIR data principles include Findability, Accessibility, Interoperability and Reusability. These will be used as guiding principles throughout this chapter. An overview of how the principles are satisfied can be found at the end of this chapter.

Requirements

The first step in creating a schema to represent governmental publications and documents is identifying requirements that will help in constructing a solid data model and detecting what particular data to store. This can be achieved in two ways: by analyzing characteristics of the real-world publications and documents, leading to more technical requirements, and by developing an inventory of end-user specific requirements.

Technical requirements

A suitable data model should fit the data it is storing. In order to identify a compatible data model, the type of data that is dealt with needs to be researched. A few characteristics can be identified. Firstly, the documents and their overarching 'publication' can be thought of as objects. A publication contains document objects. Both the publication and documents need to have attributes, such as a title and a date, which do not have to be ordered. By representing publications and documents as objects, a more understandable view of the real-world equivalents is created. Secondly, not all objects have the same relevant attributes. An active disclosure of information does not have the need to store the same data as decisions on information requests, for example a decision date or if the request is adjourned. However, both active disclosures and decisions on information requests are grouped as publications. Similarly, a released document has more relevant attributes than a decision or inventory document. Metadata on released documents contain for example a originator and recipient, and sometimes grounds of refusal according to a relevant Wob- or Woo-article. Nevertheless, all types of documents are grouped together. A third characteristic is that there is no fixed number of documents belonging to a publication. Some publications contain tens or even hundreds of documents, while other publications are refused according to a relevant ground, resulting in the sole existence of a decision document. Likewise, some attributes, like grounds of refusal, can contain more values.

End-user requirements

Also, a more user-centered approach can be taken in finding an optimal data model. For creators of the metadata, it is convenient that the data are human-readable. This will ideally improve archiving and reuse of the publications, as well as the communication between the government and it's inhabitants. For end-users of the data, like researchers, political figures and journalists, the data also must be human-readable, referring back to the experiment by van Oostveen & van Loenen (2014), which concluded that the responses to Dutch information requests were too mixed to make sense of. Also, the data must be computer-readable, as the current publications lack computer-readability. This should improve the researchability and transparency of the published information. Research trying to monitor the government on for example response time of the information requests, carried out by the Instituut Maatschappelijke Innovatie and Open State Foundation (2022), was seriously impeded by the lack of computer-readable data. About 1000 information request were analyzed by hand, producing about 12.000 records of data. Such time-consuming data gathering would be obsolete if there was suitable metadata on the publications available. More computer-readable data also provide better searchability. To improve communication, reliability and prevent ambiguity of the data, type

and value constraints based on standards, such as ISO standards, can help in creating an uniform way of storing values.

Requirement	Effects
The data model needs to be human-readable	Improves archiving and reuse of publications, improves vertical communication between the government and inhabitants
The data model needs to be computer-readable	Improves transparency and researchability, improves monitoring of the governmental activities, improves searchability and reusability
The data model needs to be able to represent publications and documents as objects with attributes	Creates a representative and understandable view of real publications and documents
The data model needs to be able to deal with changing attributes, depending on the object's type	Improves flexibility of storing only relevant attributes for an object, while not every attribute is relevant for all publications and documents
The data model needs to be able to deal with a variable number of documents linked to a publication	A single publication may contain tens or hundreds of documents
The data model needs to have certain type and value constraints	Improves uniformity, improves reusability, coordination and cooperability of governmental bodies, prevents ambiguity

Table 1. Identified requirements for a data model.

A semi-structured approach

With the requirements listed, a data model can be identified that suits the requirements best. Following the FAIR principles, to be accessible, data need a standardized communications protocol and to be interoperable, a data model needs to use a formal, accessible, shared and broadly applicable language for knowledge representation. The need for a clear, uniform data model and language for knowledge representation flows from these principles. Data models may vary on a scale from structured to unstructured. Unstructured data are untaged, unrestricted and undefined. Generally, unstructured data are not organized before storing or processing. This makes it impossible to store in a traditional database management system. Examples include log files, phone recordings, photos and chat messages. Structured data are described as the opposite. It has a well pre-defined structure and is generally stored in a relational database. From analyzing the publications and associated documents, their relations and attributes, the conclusion can be drawn that the needed data model is neither structured nor unstructured. It is not completely structured, because the model needs to deal with changing attributes and a variable number of documents attached to a publication. This makes it hard to organize the data in a tabular format, although it is possible. The drawback of this approach would be that publications and documents need to be stored in separate tables, decreasing human-readability and the level of object-representation. Also, it is not unstructured, because a number of tags or attributes can be identified to indicate and separate semantic elements of the data. Looking at the perspective of the data users, an unstructured approach would also not be a great fit, as this does not solve the problem of human- and computer-readability. There is a need for a flexible format. The answer is a *semi-structured* approach. The key idea is the representation of data as a tree-like structure (Buneman, 1997). Also, semi-structured data is said to be self-describing and suitable for object-oriented databases. This fits the challenge of storing publications and document objects, where documents belong to publications. It handles changing attributes and a variable number of documents, while still maintaining the ability to add tags and constraints to the data. The taken approach will incline towards a more structured approach, as type and value constraints on the data need to be introduced as well as predetermined, well-defined attributes. Semi-structured file formats like the Extensible Markup Language (XML) and JavaScript Object Notation (JSON) follow this tree-like

Met opmerkingen [ML1]: Introductie structured en unstructured, voorbeelden

structure, making it possible to store publications and attached documents in one single object. This adds human-readability, while maintaining computer-readability.

|JSON vs. XML|

Two well-known file formats satisfy the requirement of supporting the storage of semi-structured data. These are the Extensible Markup Language (XML) and JavaScript Object Notation (JSON). XML requires structuring the data into a document structure, JSON structures are based on arrays and records, making it a simpler data exchange format (maybe 2 examples). XML and JSON have much in common, but JSON has a smaller grammar and is based on data structures used in modern programming languages, making it better compatible with languages like Python and Javascript when analyzing the data (Crockford, 2006). A case study by Nurseitov *et al.* (2009) indicates that JSON is faster and uses fewer resources than XML. Because of the simplicity of JSON, it fits the requirement of human-readability better than XML. Being better integrated into modern programming languages, JSON complies to the requirement of researchability, making it easier to analyze data directly before having to parse all the data first. Moreover, JSON is used in the Elasticsearch stack, tackling searchability. On the other hand, the advantage of XML is the support of schema languages. These describe the structure of XML documents, making it easy to introduce restrictions and validate the correctness of the data. However, a schema language called JSON schema has been shaped to annotate and validate JSON documents, offering the same functionality as XML schema languages (Droettboom, 2022). With JSON offering more simplicity and offering the same popular functionality of XML JSON for this exact purpose is recommended. The remainder of this research will contain examples and use-cases in JSON.

Attribute selection

According to the findability and reusability guiding principles, data need to be richly described with a plurality of accurate and relevant attributes. Relevant attributes for publications and documents can be found and extracted from different points of view. These include consulting existing metadata standards on files, exploring existing journalistic solutions, using earlier work and research, analyzing provided metadata for current publications and the more scientific approach of identifying technical metadata for research purposes. This chapter will treat all points of view, going from abstract metadata from existing standards to more case specific attributes. At the end of each source, the additional relevant attributes that will be used are listed.

Metadata standard (Dublin Core)

Numerous of metadata standards and ontologies have been proposed to consistently store metadata on subjects. The Dublin Core Metadata Initiative (DCMI) created a set of core elements to describe resources, such as video, images and files. The element set has been formally standardized as ISO 15836 (International Organization for Standardization, 2019). The original version consists of fifteen metadata elements, however not all are applicable on this case. The standard aids in formulating some basic metadata on publications and documents. For publications, we can identify the following attributes: identifier, title, subject, description, date and type. The identifier attribute is in line with the Findability principle, as metadata needs to have a globally unique and persistent identifier which makes the publications accessible by their identifier. The type attribute will help in determining if the publication is based on a request or an active disclosure. For documents, the following set of attributes is extracted from the Dublin Core standard: identifier, title, description, format, date, type, language and rights. The type attribute can be used twice. Firstly to denote the document type (request, decision, inventory list or released document) and secondly, in the case of a released document, the type of released document, such as a memo, an agenda or an email. Attributes like contributor, creator, publisher and source are considered too generalized for the publications and documents, as these attributes call for more case specific metadata. This will be further elaborated in the next sections. The remaining Dublin Core attributes, coverage and relation, seem not applicable for publications and documents.

Publication: identifier, title, subject, description, date, type

Document: identifier, title, description, format, date, type, language, rights

[Journalistic standard \(Follow the Money\)](#)

Next to considering the somewhat technical Dublin Core standard, a more end-user centered approach can be taken. This approach is more focused on extracting meaningful information on the data. Dutch research journalism platform ‘Follow the Money’ (FTM) created their own Follow the Money data model to organize concepts in a for journalists useful way to investigate things (Follow the Money, 2021). The hierarchical data model contains two relevant objects with attached metadata. On top of the hierarchy is a ‘thing’, which describes a real-world object. A ‘document’ is a subtype of a ‘thing’, directly relating to this case’s released publication documents. A thing contains some of the same attributes as Dublin Core, like a name (title), a description and topics. Two new useful and relevant attributes can be added to that list: a source url (more specific than the Dublin Core ‘source’) and the ‘retrievedAt’, which can be interpreted as a download date. These attributes introduce a new kind of attributes. This new kind includes attributes that are obtained after publication and can be filled during research of the metadata, while the source url and a download date are not available when publishing the documents. Inspection of the defined ‘document’ object also yields new attributes: file size, file name and body text all contain useful information on documents. A document also contains the mime type and extension attributes, replacing Dublin Core’s ‘format’ attribute.

Publication: sourceUrl, retrievedAt

Document: sourceUrl, fileSize, fileName, bodyText, mimeType, fileExtension

[Earlier work \(Open State, VNG, North-Holland\)](#)

From research perspective, The Open State Foundation in cooperation with the province of North-Holland and the association of Dutch municipalities (VNG) already created a concept version of a guideline for standardized attributes to attach to information request, to improve publishing and archiving the information requests (Open State, VNG, Provincie Noord-Holland, 2021). The guideline, named OpenWob, contains recommendations on request level and is composed in cooperation with a province and municipalities. As these bodies will handle the information requests, the guideline contains useful attributes for handlers and creators of the information requests, as well as some request specific attributes. Firstly, some attributes are already covered by Dublin Core or FTM: identifier, title, subject (called a theme in the guideline) and a description. The guideline splits the date out in a file date and a decision date. Furthermore, the guideline provides some new attributes: the responsible body for handling the request, the requester, an internal id, a boolean attribute indicating adjournment of response, the first recipient of the request and some geographical attributes. The handler of the request of publication seems a better describing attribute than Dublin Core’s ‘author’, ‘publisher’ or ‘contributor’.

Publication: fileDate, decisionDate, handledBy, requester, internalId, adjourned, firstRecipient

[Analysis of current publicly available publications](#)

Webpages from the Dutch government, municipalities and provinces often already contain a number of publications with documents. By analyzing these publications, together with their decision documents and inventory lists, already used attributes and potential new attributes can be listed. This section will discuss three ways of identifying attributes: by analyzing the publication webpages, the decision documents and inventory lists.

[Publication pages](#)

The Dutch government usually provides a title, description and responsible body on the publication page. For documents, the title, format, number of pages and file size are provided. Also, the date is provided. However, this date is always the decision date, and not the date of the creation of the individual documents. By inspecting and downloading the individual documents, a file name and

extension can be retrieved. The title or filename of a file often describes the type of document, for example a decision document or an annex. All found attributes but the number of pages of a document are already defined by Dublin Core, FTM or the guideline by OpenState.

Decision documents

Decision documents contain the reaction of the handling body to an information request. In the case of an active disclosure, there is also a decision document explaining the released documents. A decision document often contains the file date of the request and the response date. Furthermore, the decision contains information about the possible adjournment of the response. An overall valuation of the response is given, for example public, partially public or not public. The decisions mentions the requester, however this information is mostly hidden based on privacy reasons. Again, mostly already identified attributes are found. The valuation of the publication is a newly added attribute.

Inventory lists

Not all publications contain an inventory list. This type of document lists all the released documents, together with some useful information about the documents. From 392 readable inventory lists the table inside the pdf document is extracted using the Python module 'pdfplumber'. Table 1 shows a frequency table of the ten most occurring column headers. The names of the headers are generalized, as for example 'nummer', 'nr' and 'documentnr' all mean the same. This also indicates that the reported count is the minimum count, as not all the tables were perfectly readable. The column headers are mostly in Dutch, but are translated in the table. Also, an attribute name is given to the column. By analyzing the inventory lists, five new document attributes are identified.

Table 2. Frequency table of the ten most occurring column headers of 392 inventory list tables

Column name (Dutch)	Column name (English)	Attribute name	Count
Nummer	Number	identifier	295
Documentnaam	Document name	title	278
Weigeringsgrond	Ground of refusal	groundsOfRefusal	268
Beoordeling	Valuation	valuation	260
Afzender	Originator	originator	204
Datum	Date	date	203
Ontvanger	Recipient	recipient	198
Soort	Type	annexType	82
Tijd	Time	date (generalized)	42
Vindplaats reeds openbaar	Location of already public document	alreadyPublicLocation	28

Publication: numberPages, valuation

Document: groundsOfRefusal, valuation, originator, recipient, alreadyPublicLocation

Extractable metadata

To improve monitoring research on information publication, a few easily extracted attributes can be added. From the document collection, an attribute on number of documents can be added to quickly determine the size of the publications. Also, the documents itself can be enriched with useful attributes, such as the number of pages, the number of pages with text on it, the total number of words and characters, if the documents is a scan and requires Optical Character Recognition (OCR) in order to read the document. These attributes stimulate researchability and transparency on the quality of the publications. While the information gain would not outweigh the amount of work to extract these attributes per document or publication, these values can be filled by analyzers of the data. Next to that, an easy software tool may be able to automatically extract and fill these fields by providing the

document itself. The automatic production of desirable JSON documents, with the extraction of useful additional metadata, will be covered extensively in the next chapter.

Publication: numberDocuments

Document: numberTextPages, numberWords, numberCharacters, isScan

Complying to the FAIR principles

Earlier, the FAIR principles (Findability, Accessibility, Interoperability and Reusability) were introduced. In this chapter, a complete data model with attributes, flowing from domain-specific requirements, is proposed. Table (?) shows how the data model satisfies the FAIR principles.

FAIR principle	Data model characteristics (Specific FAIR principle)
Findability	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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)

Buneman, P. (1997, May). Semistructured data. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems* (pp. 117-121).

Crockford, D. (2006). JSON: The fat-free alternative to XML. <http://www.json.org/xml.html>.

Droettboom, M. (2022). Understanding JSON Schema. Available on: <http://spacetelescope.github.io/understanding-jsonschema/UnderstandingJSONSchema.pdf>.

Follow the Money. (2021). *followthemoney documentation*. Geraadpleegd op 13 april 2022, van <https://followthemoney.readthedocs.io/en/latest/entity.html>

Instituut Maatschappelijke Innovatie & Open State Foundation. (2022). *Ondraaglijk traag – Analyse afhandeling Wob-verzoeken*.

International Organization for Standardization. (2019). DATE AND TIME FORMAT. (ISO Standard No. 8601). Retrieved from <https://www.iso.org/iso-8601-date-and-time-format.html>

International Organization for Standardization. (2019). Information and documentation — The Dublin Core metadata element set — Part 2: DCMI Properties and classes. (ISO Standard No. 15836-2). Retrieved from <https://www.iso.org/standard/71341.html>

Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: a case study. *Caine*, 9, 157-162.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3(1), 1-9.

Attribute	Dublin Core (DC)	FTM	OpenWob	Woo	Research
identifier	x		x		
title	x	x	x	x	
subject (topic, theme)	x	x	x	x	
subjectId			x (based on)		
description	x	x	x	x	
fileDate	x (date)	x (date)	x	x	
decisionDate	x (date)	x (date)	x	x	
type	x			x	
sourceUrl		x			
retrievedAt		x			
handledBy			x	x	
requester			x	x	
internalId			x		
adjourned			x	x	
firstRecipient			x		
valuation				x	
numberDocuments				x	x

Attribute	Dublin Core (DC)	FTM	OpenWob	Woo	Research
identifier	x			x	
title	x	x		x	
description	x	x			
date	x	x		x	
fileExtension	x (format)	x		x	
contentType		x			
documentType	x (type)			x	
language	x	x			
rights	x				
sourceUrl		x			
fileSize		x		x	
fileName		x		x	
bodyText		x			
annexType	x (type)			x	
valuation				x	
groundsOfRefusal				x	
alreadyPublicLocation				x	
originator				x	
recipient				x	
numberPages					x
numberWords					x
isScan					x
numberTextPages					x
numberCharacters					x