Initial Specification of Enrichment Tools

Boer, A.; Winkels, R.

Publication date
2015

Document Version
Final published version

License
CC BY-SA

Link to publication

Citation for published version (APA):
Boer, A., & Winkels, R. (2015). *Initial Specification of Enrichment Tools*. Universiteit van Amsterdam.

JUST/2013/ACTION GRANTS

Grant Agreement Number 4562

Project Start date: 01.04.2014

Project End date: 31.03.2016

Deliverable 2.2.d2 – Initial Specification of Enrichment Tools

Version: 1.0

Document prepared by: UNIVERSITEIT VAN AMSTERDAM
Alexander Boer, Radboud Winkels

Contributors:

Deliverable due date: 31.1.2015

Deliverable actual date: 25.2.2015

co-funded by the Civil Justice Programme
of the European Union

Document History

Date	Revision	Comments
01.12.2014	0.1	creation
08.01.2015	0.2	Section 4 added
05.02.2015	0.3	Section 5 and 6 added
25.02.2015	1.0	Final revision

Document Authors

Alexander Boer, Radboud Winkels

Participant List

	Short Name	Organisation Name	Country
1	UVA	Universiteit van Amsterdam	NL
2	SUSS	University of Sussex	GB
3	LSE	London School of Economics and Political Science	GB
4	ALP	Alpenite srl	IT
5	SUAS	Fachhochschule Salzburg GmbH	AT
6	BYW	BY WASS GmbH	AT

Disclaimer:

This publication has been produced with the financial support of the Civil Justice Programme of the European Union. The contents of this publication are the sole responsibility of *University of Amsterdam* and can in no way be taken to reflect the views of the European Commission."

Executive Summary

This report specifies the enrichment tools in the Openlaws.eu project, and follows up on report 2.2.d1 (requirements of enrichment tools). Openlaws.eu aims to initiate a platform and develop a vision for Big Open Legal Data (BOLD): an open framework for legislation, case law, and legal literature from across Europe.

It describes that nature of BOLD objects, peculiarities of legal documents and functionalities for users of OpenLaws.eu to manipulate, search and share them.

Enrichment can happen two ways basically: Either by humans ('wisdom of the crowd') or automatically. Both methods will lead to additional metadata that will be stored with the original legal data.

1 Table of Contents

1 Introduction

Openlaws.eu aims to initiate a platform and develop a vision for Big Open Legal Data (BOLD): an open framework for legislation, case law, and legal literature from across Europe. Based on open data, open source software and open innovation principles we are adding a *social layer* to existing *legal information* systems. This document follows up on report 2.2.d1 (requirements of enrichment tools, and specifies the BOLD enrichment tools.

This document primarily follows the familiar model-view-controller paradigm for interactive aspects of the enrichment tools, separating:

1. *Models* of BOLD objects
2. *Views* on BOLD object models
3. *Controllers* of BOLD object models, and
4. *Pipelines* related to BOLD object models.

Section 2 starts with a specification of models of the objects that make up a Big Open Legal Data (BOLD) framework, in the present document called the *BOLD objects*, addressing both big open legal data and the envisioned social networks that will keep the process of enrichment going.

Having introduced these, several sections follow that specify model enrichment processing pipelines, and views and controllers for the BOLD models identified.

2 Models of BOLD objects and BOLD networks

BOLD objects are identified by one or more URIs (incl. URL, URN, etc), originating from different identification schemes.

Legal data, narrowly conceived, consists of four major types of BOLD objects:

1. Documents are structured texts, hierarchically decomposable into linked lists of document fragments;
2. Metadada about documents and document fragments:
   a. references are labeled links between document fragments, and
   b. labeled links can be used to create arbitrarily complex features of documents and document fragments;
3. Folders and clipboard containing documents, document fragments and metadata;
4. Groups of users.

Enrichment tools functionality is closely linked to BOLD object type, specifically:

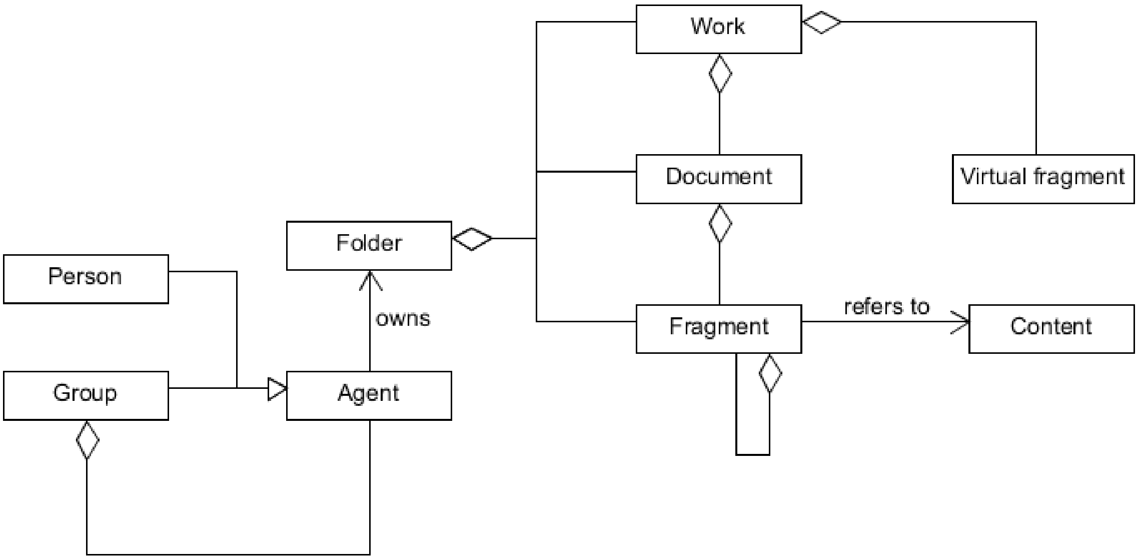
1. Content object

2. Set-decomposable object

3. Linked-list-decomposable object

4. Document
5. Document fragment
6. Labeled link object
7. Graph
8. Reference
9. Folder
10. Clipboard
11. Group
12. Person

An overview of the main object types and relationships is given in figure [..].



Documents and content

***Content* objects** contain data, and some, but not necessarily all, content URI permit dereferencing to retrieve this data. Content objects are distinguishable by type depending on the expected structure of the data, the operational semantics associated with that structure, and method for dereferencing.

Generally, BOLD objects are subdivided into :

1. *simple objects,*

2. objects that are decomposable into a *set of BOLD objects*, and
3. objects that – besides being decomposable into sets of objects – are decomposable into linked lists, or totally ordered sets, *of BOLD objects* of objects.

BOLD objects may participate in multiple decompositions. Wherever sets of objects are serialized:

Linked lists of *content* objects may be lexically structured in order-preserving XML/HTML/PDF data structures. HTML is strongly preferred.

Unordered sets of BOLD objects are, when not embedded in input XML/HTML/PDF data structures, lexically structured in RDF or JSON.

A **BOLD document** is a content object that can be decomposed into a linked list of **BOLD document fragments**. A BOLD document fragment is a content object that *may be* decomposable into a linked list of document fragments. When conceptualizing the document as a tree, all leafs of that tree are content objects. The leafs are the smallest level of structural subdivision that is (in that type of document) commonly *referred* to with an unambiguous reference. Below that level the leaf document fragment *content* may be further marked up, but this does not count as BOLD object decomposability.

Note that structured texts may require alternative structural decompositions, **but that prototypes of the encoding tools may not support this**:

1. Text structured into chapters and articles may have an alternative decomposition into pages, with footnotes; and
2. in the annotation of individual sentences with markup in HTML with SPAN elements, annotators may come to alternative structural decompositions of a sentence, depending on purpose of the annotation.

User-created folders and metadata

Labeled links between BOLD objects are *(subject predicate object)* triples. Triples are a type of simple BOLD objects. RDF and JSON data has a standard interpretation as *(subject predicate object)* triples, forming a graph. These triples may be characterized where appropriate by:

1. a subset *R* of the possible subjects, predicates, and objects (product S X P X O),
2. a set of *edges* with predicate *p* E(p) = { (s,o): (s, p, o) in *R*}, or
3. a set of *features* of a subject *s,* F(s) = { (p,o): (s, P, o) in *R*}.

BOLD graphs are BOLD objects decomposable into a set of labeled links between BOLD objects. Graphs are the raw material for application of network analysis techniques.

BOLD references are labeled links between document fragments.

Folders and clipboards are user-created editable unordered sets and linked lists of BOLD objects, respectively, mainly initiated by user search and copying actions.

Social networks

Social network graphs are composed of persons and groups, both types of agent.

Groups are decomposable into sets of agents, shared memberships of a group can be interpreted as links between agents and vice versa, and shared members can be interpreted as links between groups and vice versa, following a social theory attributed mainly to Breiger [C]. Persons are agent that are not decomposable (i.e. leaf agents). Persons may play distinct roles[1] that can be interpreted as group memberships (and vice versa). Groups may be directly expressed by persons as a creative act, or discovered through network analysis techniques by looking at their activities as producers and users of BOLD. Persons may decide to which group of which they are member productions and uses are associated, and they may constrain membership of groups or visibility of documents to specific agents.

3 Bibliographic identity

Bibliographic convention [D] is to distinguish legal texts and text fragments on at least four levels, as distinguished by MetaLex [B]:

1. On the *item level* legal texts and text fragments can be dereferenced by identifier and copied, resulting in a new item;
2. On the *manifestation level* any change to the data produces a new manifestation, including a change of data format, annotation of structure, or the embedding of metadata;
3. On the *expression level* only a change of the text by its author produces a new expression;
4. On the *work level* a text is identified by the details of its publication: as long as the title, author, and publication date remain the same, expressions are versions of the same work.

The **BOLD document** is essentially a manifestation. Most of the metadata refers to the corresponding work or expression, however. **In a prototype of the enrichment tools, only one manifestation of each expression may be assumed to exist.**

A **BOLD work may be decomposed into a set of expressions**. A BOLD document may be decomposed into a set of manifestations. Enrichment consists of 1) creating alternative manifestations of an expression, or 2) adding metadata about an expression or work. One work may be expressed multiple times. An expression may have many manifestations. Items may be freely copied, resulting in new items.

Versioning

For regulatory text, the distinction between works and expressions is of critical importance, because the text is typically changed over time. Most works are decomposable into a single linked list of expressions over time. In some cases (retroactive annulment of changes and unforeseen changes to scheduled changes in the future) the versioning chain may change over time retroactively, resulting in alternative versions of a text, in which case the versions cannot be expressed as a linked list. **In a prototype of the encoding tools a single linked list of**

[1] E.g. a legal scholar specialized in insolvency may at the same time be a part time judge in a cantonal court mainly dealing small claims, and has distinct information needs within these distinct capacities, manifested in his production and use of information.

expressions may be assumed to exist.

Languages

Many works are moreover available in alternative language variants. Support of this is obviously an important requirement in the EU. Each language variant is a variant of another expression.

References

Conventionally, regulatory text refers to other regulatory text on the work level: which version should be used is left to the reader. Some call this a *dynamic* reference. A court decision refers to a specific version of a regulatory text on the expression level. Some call this a *static* reference. Any other text that refers to legislation by default refers to a specific version, unless the text is under editorial control and guaranteed to be up to date with the text it refers to. Obviously, texts may discuss an old version, compare an old version to a new version, compare two language variants of a version, or (very frequently) discuss an anticipated version of a regulatory text[2].

Mixed content and quoting

Legal text is often quoted, in modifying legislation, in court decisions, papers and books, etc. Quoting is an alternative way of referencing information, and the quoted text fragment is both a part of the quoted and of the quoting document. It is moreover a potential source of interesting and innovative manifestations of text fragments.

3.3 Folders, shopping carts, and shopping lists

Users looking for legal information require some way to select and keep it. One may think of this in terms of the shopping cart metaphor. By storing the contents of a shopping cart in a folder the user enriches the text fragments in the shopping cart with features. The folder is a set based on shared features.

Alternatively, the user may use the shopping cart as a basis for producing a new text, quoting the selected text fragments. In a new text the fragments are ordered in a chain; the shopping list, whose order may be changed, is a more appropriate metaphor for this use case.

Events and other mediating objects

The ultimate purpose of enrichment is to uncover the contexts in which legal data is used and produced. To accurately describe these, user communities should be able to freely introduce entities such as events, business processes, services, logical rules etc. as long as these can be identified by URI. The use case for event-based metadata was discussed in detail in [A].

Audio and video data

Since OpenLaws.eu will develop app-based support for BOLD, it makes sense to consider audio and video-based legal data. On a smart phone, an audio recording is easier to make than

[2] The news value of a discussion of a legal rule is highest well before the new rule goes into actual effect. The importance of anticipation of changes in the law should not be underestimated!

a text. Disadvantage of this mode of annotation is its lack of accessibility for most automated enrichment tools, although it may still play a useful role for network analysis.

3.4 MetaLex conformance

OpenLaws will be based on conformance with the MetaLex standard for legislative documents. The design principles for BOLD objects identified thus far are therefore inspired to a large extent by MetaLex.

OpenLaws does not cover just legislation, but other sources of legal information as well, besides user-managed shopping lists and folders. These documents and folders will 1) refer to legislation, meaning that the references should meet the requirements, and 2) literally quote from (or include) legislation, meaning that they embed alternative manifestations of a part (a article, a sentence, etc) of the legislation.

Implied design requirements

In this section we list MetaLex-based requirements briefly, for convenience of the reader. Full requirements are found in the specification[3]. The main design principles of MetaLex [B] are the following:

1. Legislative documents, and their parts, can be individuated on the work, expression, manifestation, and item levels of abstraction.
2. The HTML or XML structure is a manifestation of a document and its parts. The structure of documents can be described without ambiguity using a limited number of content models.
3. Work, expression, and manifestation, and their parts, should have a unique name that is an IRI reference.
4. These names should meet the requirements of some naming convention. A transparent (meaningful) name may be interpreted as a set of identifying metadata, and vice versa, a set of identifying metadata may be associated to an opaque (meaningless) name instead.
5. Metadata in general is about the document as a work, as an expression, or as a manifestation. It uses the right IRI reference as a subject.
6. Metadata can be interpreted as RDF triples.
7. Most legislative metadata describes events that happened (or will happen) with the document (as a work, expression, or manifestation)[4].
8. References to legislative documents made by authors usually refer to the work (dynamic reference), or to the expression (static reference), and not to the manifestation or item level. A correct technical reference is not the same as a direct hyperlink.

Note that it is possible to make XHTML documents conform to this standard. Another

[3] ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA15710-2010-Metalex2.pdf

[4] Example of points 5 and 7: a last-modified property with a date means different things depending on whether it applies to 1) the expression (the actual text was modified by its author, the legislator) or 2) the manifestation (the XML or HTML markup was modified by an editor, but the text remained the same). Moreover, the date is not a direct property of the document, but of an event (a modification) that either 1) happened to the work, and resulted in the expression, or 2) happened to the expression, and resulted in the manifestation.

important source of possible input is Akoma Ntoso XML, which is known to conform to MetaLex requirements.

Transforming existing text

To transform existing documents from other corpuses, consider:

1. The mapping to MetaLex content models
2. Decide on a naming convention. Do existing document identifiers and metadata give enough information to distinguish and name works, expressions, and manifestations? How much of the version history of the documents is available? Are the documents multi-version manifestations, that should be cut up and replicated into multiple documents?
3. Reinterpret embedded metadata as event descriptions, described by RDF triples.
4. Resolve references, either by looking at link metadata, or IRI references, or the text of the reference itself. Do existing references make clear whether work, expression, or manifestation are the intended targets of a reference?

Local replication of documents and metadata

1. During the project, documents will be replicated in the OpenLaws infrastructure[5]: although there are good reasons not to replicate data, replication will make development of tools easier.
2. Since the repository(y/ies) should be up to date, incremental, non-destructive updates should be made automatically[6].
3. Metadata will be stored/replicated in a dedicated repository, instead of depending on it being embedded in the (manifestations of) documents, to make development of tools easier.
4. Embedded metadata in documents viewed by users duplicating metadata in the repository may exist, but is not directly accessed by tools.
5. Users should be able to export documents from the OpenLaws infrastructure for purposes of a) printing and b) archiving self-contained and self-describing documents. For this second purpose, retaining original metadata, or even adding inserting newly created metadata from the repository or user folders, is important.
6. Note that user-created metadata should also be considered to be stored in a user-created folder, however, and user-created metadata marked as private is not shared with the repository.

Meeting requirement 2 for all corpuses of text included in the OpenLaws infrastructure may not be realistically feasible in the project. Within the project, it may not be realistically feasible to embed a concise metadata description of a document in the document based on the

[5] similar to the approach of example server http://doc.metalex.eu

[6] The source code of the updating script of example server http://doc.metalex.eu is freely available as an example.

entire metadata repository[7], as implied by requirement 5.

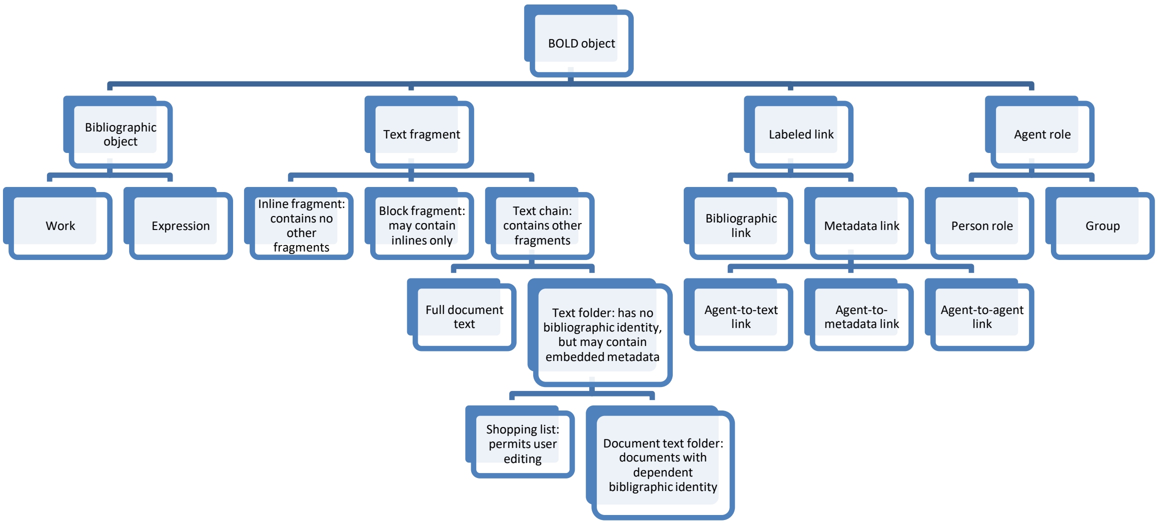
4 Classes of BOLD objects

The figure below presents a taxonomy of BOLD objects. For bibliographic objects and text fragments, additional views can be found in the MetaLex OWL schema. Note that texts are manifestation-level entities, although not all texts may be considered true bibliographic objects. Texts that are not part of a full document are not bibliographic objects.

The enrichment tools do not embed metadata into documents, and do not take account of any metadata present in documents. Metadata is embedded in folders made by users and automated enrichment tools. The shopping list is a mediating object that permits user editing before it is saved as a folder in another folder. As the metaphor implies, only one shopping list is active while a user is browsing for relevant BOLD objects.

Dependent bibliographic identity may for instance occur for treaties and associated authorized translations and protocols, acts of parliament with associated explanatory memorandum or authorized translations, etc.

The other object types are self-explanatory.


BOLD object
Bibliographic object
Text fragment
Labeled link
Agent role
Work
Expression
Inline fragment: contains no other fragments
Block fragment: may contain inlines only
Text chain: contains other fragments
Bibliographic link
Metadata link
Person role
Group
Full document text
Text folder: has no bibliographic identity, but may contain embedded metadata
Agent-to-text link
Agent-to-metadata link
Agent-to-agent link
Shopping list: permits user editing
Document text folder: documents with dependent bibligraphic identity

5 Accessing documents and texts

5.1 Requirements for text fragments

The text fragment is the fundamental unit of the user interface: (named) containers, like article 1, paragraph-sized blocks, or sentences.

[7] Depending on the structure of the repository and the logical inference involved, this may be a prohibitively hard problem.

Views

1. Displaying a text fragment, in a resizable view, with an appropriate layout.
2. Displaying a text fragment chopped to a specific character length, or until the fragment meets certain information requirements.
3. Displaying a globally correct citation of a text fragment, on the expression level, including date of last revision or publication and a title[8].
4. Displaying a *locally* correct citation of a text fragment, relative to the text of which it is part, not repeating elements already obvious[9].
5. Drawing attention to inline annotations to permit manipulations.

Manipulations

1. Folding or unfolding text fragments to show more or less information.
2. Selecting a text fragment.
3. Adding the text fragment to the shopping cart.
4. Selecting an inline annotation in a selected text fragment, and performing the appropriate action.
5. Accessing alternative known manifestations of the text fragment.
6. Accessing the previous and next expression-level version of the text fragment.
7. Accessing alternative language variants of a text fragment.
8. Accessing closely related texts[10].

Process and provenance aspects

1. Correct citation style depends on the type of document. To access the information for a globally correct citation one should have access to the entire text. In some cases the ability to select alternative styles may be required[11]. Coverage of all relevant styles is not realistically feasible in the project.
2. Appropriate layout may depend on the type of document.
3. Known manifestations of a text fragment include those that only exist in a user folder, or as quoted content of another text.

5.2 Requirements for text fragments in a shopping list or shopping cart

In addition to all other requirements for text fragments.

[8] For instance *LJN AI5638* is a correct case law identifier, but not informative to the reader. *Gerechtshof 's-Gravenhage 4 september 2003, LJN AI5638 (Scientology)* is perfect.

[9] For instance *article 1*.

[10] Incoming and outgoing references, shared features, texts that quote it, etc.

[11] The ECLI for instance competes with state-specific conventions like the example in footnote 5. For academic article and book citation styles existing solutions may be available.

Manipulations

1. Creating inline annotations[12] in a selected text fragment[13].
2. Adding features to the text fragment.

Process and provenance aspects

1. As required by basic XML/HTML, annotations in a *single manifestation* of a text fragment should not overlap.

5.3 Requirements for chains of text fragments

Applies to any text fragment that can be broken into a chain of text fragments, including full texts and shopping lists. All requirements of text fragments apply, except that the chain of text fragments is not chopped to a specific length, but is applied to its parts.

Views

1. Chopping (long) text fragments to a specific character length, or until the fragment meets certain information requirements, to show more of the text in the screen.

5.4 Requirements for shopping lists and shopping carts

Shopping lists are chains of text fragments that can be edited, to create a new text[14]. Other requirements apply. A shopping cart is a shopping list that has not been explicitly ordered or edited (yet).

Manipulations

1. Reordering text fragments in the list.
2. Adding features to the shopping list.
3. Writing a new text fragment and inserting it in the place of choice in the shopping list.
4. In an app platform for smart phones: recording audio or video and inserting it in the place of choice in the shopping list.

6 Enrichment tools

6.1 Users working with shopping lists

Views

1. Are ordered sets, and may be reordered: think of them as newly created hypertexts.
2. May include metadata (triples), text fragments and entire documents, and audio and video recordings.

[12] This requirement is very important, but may be prohibitively hard to realize in the project.

[13] Because a new manifestation of the text fragment is created that does not replace the one which was edited.

[14] The interaction model we have in mind is that of for instance *Storify*.

3. Any texts or text fragments are included in the shopping list *qua* work, expression or manifestation.

Manipulations

1. The user drops resources into the shopping list while browsing, or creates a text or audio or video recording himself.
2. May be reordered.
3. Allows annotation of text fragments with XML tags (qua manifestation), resulting in a new manifestation of that text fragment that replaces the original.
4. Can be saved as a user-managed folder.

Process and provenance aspects

1. It is possible to treat the shopping list and user-managed folder as a MetaLex XML file.
2. Membership by reference (href), by inclusion (src) should be distinguished.

6.2 Users working with user-defined folders

Views

1. The folders view models file explorer widgets.
2. Folders may be associated with descriptive metadata.

Manipulations

1. The user-defined folder is created 1) as a new folder or 2) by saving a shopping list in a parent folder and giving it a title, and descriptive metadata.
2. Folders may be made (in)visible to other groups.

6.3 Users managing and groups and roles

Views

1. Hierarchical view like the folder view

Manipulations

1. Creating groups and roles
2. Adding yourself (role) to an open access group
3. Requesting (role) invitation to a group
4. Inviting others by role to a group you are member of
5. Switching active role for creating a shopping list

6.4 Search and recommendations

Views

1. Corpuses are hierarchically laid out as folders, and filters can be applied to them. Visible user-defined folders can be searched as well.
2. Search results are displayed as chains of text fragments.

Manipulations

1. Search a corpus with keywords, optionally a reference date (expression level or work level search), and a granularity setting (size/type of the text fragment in which the keywords should occur).
2. Search results can be saved as a user-defined folder, and the search can be rerun on the folder.
3. Recommendation of search results: text fragments in search results can be reordered, and hidden from view by the user.
4. Recommendation of user-defined folders.
5. Recommendation of users/roles/groups.

Process and provenance aspects

1. Should keep a full log of recommendation decisions, including the decision-making users/roles. This is important for improving network analysis over time.

6.5 Network analysis functions

Views

1. For any text fragment, alternative expressions and manifestations of that text fragment;
2. All incoming and outgoing references, and co-referenced text fragments[15];
3. All metadata that has the text fragment as subject or object;
4. All folders of which the text fragment is a member; and
5. Any list of results should be ranked based on recommendation.

Process and provenance aspects

1. The OpenLaws infrastructure should make all relevant network data available for daily download in packages.
2. The frequency with which network analyses are run should be clear to the user.
3. All network data and recommendations have an author, being the author of the document from which data was taken, an OpenLaws user, or a network analysis algorithm. Determining the confidence in authors always has a high priority.
4. One recommendation process randomly assigns recommendations, and monitors changes, for validation purposes.

[15] Text fragments that are referenced by the text fragment that references the focus text fragment.

6.6 The OpenLaws.eu production pipeline

Although the OpenLaws demonstrator will use a single server instance, the concept of an open infrastructure implies a decentralized infrastructure, not managed by a single information owner. Synchronization of legal information is therefore not guaranteed for the future, and network analysis functions should not depend on the assumption that all legal information is always synchronized. A certain *core subset* of metadata, extracted from documents as they are passed through the pipeline, is however shared between all instances of OpenLaws.eu.

Process and provenance aspects

1. Certain network analysis functions belong to the core functionality of the OpenLaws.eu functionality: all instances, if using the same version of all components, will extract the same metadata from a document.

2. The operative principle for core functionality is (test-retest and inter-rater) reliability. If there are differences of opinion about whether a function meets this standard, it does not.

3. To the core functions belong: inverted indices for keyword search, and derivative metadata, publication metadata, and metadata on the resolution of references between documents.

References

A. A. Boer. Using event descriptions for metadata about legal documents. In: R. Winkels and E. Francesconi, editors, *Electronic Proceedings of the Workshop on Standards for Legislative XML, in conjunction with Jurix 2007*, 2007.

B. A. Boer and T. van Engers. A MetaLex and metadata primer: Concepts, use, and implementation. In Legislative XML for the Semantic Web, pages 131–149. Springer, 2011.

C. R.L. Breiger. The duality of persons and groups. In: B. Wellman, S. Berkowitz (Eds.), Social Structures: Network approach, Cambridge Univ. Press, Cambridge (1988), pp. 83–98.

D. K. G. Saur. Functional requirements for bibliographic records. *UBCIM Publications - IFLA Section on Cataloguing*, 19, 1998.