GDPR Annotation
Semantic Annotation Guide

François Lévy, Adeline Nazarenko

Table of contents

Introduction

Objective

The semantic annotation of the GDPR aims to facilitate the exploration of the text for the lawyer and to quickly identify the provisions related to a given topic, related to a particular actor or related to another provision. The provisions are segmented into one or more fragments of text (or elementary provisions) and the annotation consists in 1) associating a type to those fragments, 2) identifying the key entities which play a role in them and 3) making the most important inter-fragment relationships explicit.

The objective is to facilitate the analysis and interpretation work of legal experts by allowing them to explore the text. Using queries combining textual patterns and semantic key elements associated to the text with the annotations, one can find a list of relevant textual fragments, each one being placed in its structural and semantic context. Other navigation or exploration devices are also possible.

The annotation is neutral in that it is not intended to interpret the text. It is limited to formalizing semantic elements or relations which are explicit in the original text or at least are neither ambiguous nor subject to interpretation.

The semantic annotation completes the structuring of the regulation which is already organized in sections, chapters, articles, paragraphs, by making explicit the semantic structure of the provisions which compose it. To avoid redundancy, it does not duplicate the semantics already present in the structure of the document, either in the division of the elements or in their sequence, and which will be obvious to the reader anyway.

Annotation example

In concrete terms, annotation consists of taking the elementary provisions pre-annotated as fragments in the XML file (see an example on Figure 1) and annotating them semantically (see the corresponding example on Figure 2). The XML editor assists the annotator, saving time and avoiding mistakes.

```
        <PARAG IDENTIFIER="006.002">
          <NO.PARAG>2.</NO.PARAG> <leg:FRAGMENT IDENTIFIER="006.002.001">Member
States may maintain or introduce more specific provisions to adapt the application
of the rules of this Regulation with regard to processing for compliance with
points (c) and (e) of paragraph 1 by determining more precisely specific
requirements for the processing and other measures to ensure lawful and fair
processing including for other specific processing situations as provided for in
Chapter IX.</leg:FRAGMENT> </PARAG>
```

Figure 1. *Pre-annotated version of Article 6, Paragraph 2*

```
          <NO.PARAG>2.</NO.PARAG> <leg:POWER IDENTIFIER="006.002.001"
type="ruling" bearer="le_MS">Member States may maintain or introduce more specific
provisions to adapt the application of the rules of this Regulation with regard to
processing for compliance with points (c) and (e) of paragraph 1 by determining
more precisely specific requirements for the processing and other measures to
ensure lawful and fair processing including for other specific processing
situations as provided for in Chapter IX.</leg:POWER> </PARAG>
```

Figure 1. *Annotated version of Article 6, Paragraph 2*

Semantic annotation

The semantic annotation consists in associating elements of a semantic vocabulary to the pieces of text identified in the pre-annotation phase..

Semantic mark-up

The semantic mark-up of GDPR focuses on three types of information:

- The **fragments** which are sentences or pieces of sentences that make up the provisions: the annotation delimits the fragments and types them as, for example, obligations, prohibitions, powers, procedures or simple legal clarifications. There are different sorts of fragments:
    - The **autonomous fragments**, which have an autonomous type and can be interpreted independently of other fragments.
    - The **subordinate fragments**, which have a relational type and can only be interpreted in relation to another fragment.
    - The **sub-fragments,** which are portions of sentences or fragments that play a particular role in the interpretation of the fragment in which they appear.
- The **entities** that play an important role in the regulation: they are recorded in dictionaries and linked to the provisions where they play a key role.
- Two types of **relationships**:
    - Inter-fragment **relations** – notably exception relations or complement relations that link a procedure or modality to the provision that it specifies –, which generalize to relations between fragments and larger textual elements (*e.g.* paragraphs, articles);
    - **roles** that link a fragment to an entity, to indicate who has an obligation, who has a right or power, etc.

Semantic vocabulary

Here are the semantic vocabulary elements to be used in semantic annotation. Each element is described in detail in the rest of the guide with a focus on how to use it for annotation.

One can navigate through the guide either by clicking on the link in the following table or via the general table of contents.

Fragments	Autonomous fragments	OBLIGATION
		PROHIBITION
		PERMISSION
		RIGHT
		POWER *Subtypes* executive ruling
		QUALITY *Subtypes* Competence qualification responsability
		DEFINITION
	Subordinate fragments	COMPLEMENT *Subtypes* impact procedure text_specification validity
		EXCEPTION
	Sub-fragments	EXCEPT
Entities	Actors	PERSON
		LEGAL_ENTITY
	Concepts	CONCEPT
Relations	Inter-fragment relations	rel
		except
	Roles	obj
		bearer
		target

In addition to this semantic vocabulary, one element and some attributes elements are introduced to manage formal constraints :

- Temporary neutral fragments (`FRAGMENT`) are introduced at the pre-annotation step to allow annotators to identify the fragments to be semantically annotated ; they should be all eliminated during the annotation work.
- Several specific attributes are introduced to handle lists that are split in different structural elements and cannot be annotated as a unique semantic element (See Other attributes for details). These attributes are introduced at the pre-annotation step and should not be modified by annotators . [1]

Annotation coverage

The goal of the annotation is that all the articles of the regulation be split into annotated fragments but the coverage of the annotation is exhaustive neither for the entities nor for the relations:

- Only the key entities of the regulation that play a role in a fragment need to be identified.
- The fragments are caught in a complex set of relationships, but most of them are marked by the structure of the document and the order of the provisions (order of articles in sections, paragraphs and sentences within articles). The annotation focuses on other semantic relations, which are not immediately accessible to readers, especially the exceptions and complementary relations.

Annotation in practice

This section provides the minimum background for annotating a regulation. More detailed technical explanations are given in the appendix (Appendix 2: Technical details).

General notions on XML

The annotations are inserted in the text according to the XML standard. To indicate that a text segment is of type xxx, one must surround it by two **tags** indicating its type: the start tag `<xxx>` and the end tag `</xxx>`. The segment and its tags constitute an **element**. The start tag can also contain **attributes** that have a name and a value, for example `<xxx yyy="v1" zzz="v2">` (the element is of type `xxx`, has an attribute `yyy` whose value is `v1` and an attribute `zzz` with `v2` as value). An **identifier** is an attribute whose value is unique and can be used for identification. The allowed **content** of an element (what is between the start and end tags) is generally a mixture of text and other elements (called a **mixed content**).[2] .

[1] Apart from a rare and specific case described in List relationships.

[2] There are four types of content for an element in XML : empty, simple (only text), complex (with sub-elements) or mixed (with both text and sub-elements).

Except for the fact that two elements cannot overlap (they are either separate or included in each other), the standard is very flexible. Each application can specify its own constraints. The specific GDPR annotation system is presented in the following, in the form of patterns as shown in the next section.

Practical details

The annotations are added in the file `Light_GDPR_EN_xxxx.xml`[3], where a simplified version of the text structure appears in the form of structural annotations. The annotator is not expected to modify the existing structural annotations, which are part of the reference text, but to add a semantic annotation layer[4].

It is advisable to use an XML editing tool that facilitates the insertion of tags and checks their compliance with the constraints. For it to work properly, all the essential files must be in the same directory (see Provided files and the separate *Annotation Technical Guide*).

Articulation of structural and semantic annotations

To ease the annotation work, the annotators are provided with a semantically pre-annotated file, in which the boundaries of the fragments to annotate are already marked-up and a selection of key entities is already recorded in the dictionaries.

The main elements of the text structure annotation are the following:

- The provisions form the content of the element `ENACTING.TERMS`.
- Each article is being marked up as an element `ARTICLE` whose attribute `IDENTIFIER` is unique.
- Most articles are themselves divided in `PARAG` elements, which also have a unique `IDENTIFIER` attribute.
- The paragraphs are mainly composed of text but they can contain P and LIST elements.

The semantic annotations are to be inserted in the text of the paragraphs (or articles without paragraph) and must respect some constraints:

- The fragments are pre-annotated and they can only appear in the content of a structural element `ARTICLE`, `P` or `PARAG`.
- A sub-fragment can only be marked in the content of a fragment.

[3] Xxxx is a variable part of the name, according the subtask in concern

[4] Unlike the text structure annotations, the semantic ones start with a `leg:` prefix.

Constraint description

The constraints related to the semantic elements are described in the corresponding sections below in the form of patterns built on the following model:

<leg:*ELEMENT_TYPE* (*attribute_name*="(*value*|*list_of_values*)")* (*attribute_name*="(*value*|*list_of_values*)")***>*Content***
</leg:ELEMENT_TYPE>

- What is in black must be copied as it is by the annotator, what is in red must be replaced for each specific annotation;
- What is in bold is mandatory, what is in medium is optional;
- The blue characters are meta-characters that must not be copied: they indicate either a choice (..|..) ; or a possible repetition: (...)+ for 1 or more copies, (...)* for 0 or more copies.

The generic pattern above indicates that an element of the semantic layer must have a type prefixed by `leg:`, that it has a possibly empty list of mandatory attributes and another possibly empty list of optional attributes, as well as a content. The attributes have either a single value or a list of values.

Dictionaries

The dictionaries that contain the entities to which it is possible to refer in the annotation are given as separate files referred as included files in the Light_GDPR_EN_toAnnotate.xml file..

Annotation of fragments

The goal of the semantic annotation is to describe the provisions of the GDPR as a sequence of semantic fragments: each sentence is by default a fragment[5] and each fragment is associated with a semantic type, which can be semantically autonomous (*e.g.* obligations) or subordinate. In the latter case, the fragment is necessarily interpreted in relation to another fragment: for example an exception is understood only with regard to a general rule. To this are added the sub-fragments, included in a fragment and whose dependence is therefore obvious.

Like other textual elements, notably articles and paragraphs, fragments have a unique identifier (`IDENTIFIER`) which allows them to be referred to. The syntax of these identifiers is derived from those of articles (`001`, `002`, `003`, etc.) and paragraphs (eg `001.001`, `060.010`). Thus the fragment `017.003.001` is the first fragment of the 3rd paragraph of article 17.

[5] There are exceptions for provisions organised in the form of lists. (see List relationship).

In addition, a special identifier (UNDEFINED) is predefined to allow reference to undefined fragments or textual elements.

The annotators are provided with starting dictionaries files and a pre-annotated version of the GDPR in which the fragments to be annotated are already marked-up in a neutral way and associated with their unique identifiers. In concrete terms, the annotar has to read the provisions and, for each neutral fragment <leg:FRAGMENT IDENTIFIER="xxx.xxx.xxx">... ...</leg:FRAGMENT>, to

- Choose the appropriate type for the fragment (e.g. OBLIGATION, POWER) and substitute it to FRAGMENT,
- Add the relevant semantic attributes (e.g. bearer, target, rel, except) to the fragment with the appropriate values,
- In case an entity to be referred to has no predetermined identifier (id attribute) in the dictionaries, add a new entry for it in the actor or concept dictionary,
- Check if the fragment contains a sub-fragment that needs to be annotated and, if necessary, frame it with the sub-fragment tags (e.g. <leg:EXCEPT> and </leg:EXCEPT>).

The attributes already associated with the neutral fragments must be unchanged. These are the identifiers (IDENTIFIER) and attributes that are added for technical reasons (linking the parts of lists that appear in different structural elements :. is_list_header and has_list_header. See List relationships).

Autonomous fragments

OBLIGATION

Definition

An OBLIGATION is a fragment which has the semantic value of imposing an obligation on an actor: it expresses a positive legal constraint that is imposed on the actor who bears the obligation. Typical formulations are: "X must ..." or "X has an obligation to ...".

Syntax[6]

<leg: OBLIGATION IDENTIFIER = "*identifier*" bearer = "(*identifier*) +" rel = "(*identifier*) +" except = "(*identifier*) +" is_list_header = "*boolean*" has_list_header = "*identifier*"**>*Fragment mixed content*</leg: OBLIGATION>**

[6] See Practical details for the rules of interpretation of the pattern.

As with all fragments, the content of the OBLIGATION element is mixed. The text between the start and end tags may include some XML elements.

The Summary table of fragment attributes lists the attributes of the OBLIGATION element which has only 2 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the bearer role introduces the identifier of the entity that has the obligation (or identifiers if the role is fulfilled by several entities).

The fragment can also be related to other fragments or textual elements (rel and except attributes, see Dependency relationship and Exceptional relationship).

When the fragment enters a list, it may have an additional structural attribute (is_list_header or has_list_header, see List relationship).

Annotation recommendations

- The annotator has to analyze the semantic value of the fragment, bearing in mind that not all obligations contain the keyword “obligation” or the verb “must”.
- The semantics of the obligation is strong and there is generally a sanction, a penalty or a liability foreseen (not necessarily spelled out in the same text) for the actor who contravenes an obligation. This link with the sanction is looser when the actor is a legal entity.
- The value of the bearer role is often mentioned in the text content of the fragment, but this is not always the case.

Examples

```
<leg:OBLIGATION IDENTIFIER="011.002.001" bearer="p_CONT">Where, in cases referred to in paragraph 1 of this Article, the controller is able to demonstrate that it is not in a position to identify the data subject, the controller shall inform the data subject accordingly, if possible.</leg:OBLIGATION>
```

Fragment 011.002.001 is annotated as an obligation incumbent (bearer="p_CONT") on the person p_CONT (the controller), a mention of whom is made in the text content of the OBLIGATION.

```
<leg:OBLIGATION IDENTIFIER="012.003.004" bearer="p_CONT"> Where the data subject makes the request by electronic form means, the information shall be provided by electronic means where possible, <leg:EXCEPT>unless otherwise requested by the data subject</leg:EXCEPT>.</leg:OBLIGATION>
```

Fragment 012.003.004 is annotated like the previous one, as an obligation incumbent on the controller (bearer="p_CONT"), but it also includes an EXCEPT sub-fragment which signals an exception to the general obligation.

```
<leg:OBLIGATION IDENTIFIER="045.008.001" bearer="le_EC" rel="045.003
045.005">The Commission shall publish in the Official Journal of the
European Union and on its website a list of the third countries,
territories and specified sectors within a third country and
international organisations for which it has decided that an adequate
level of protection is or is no longer ensured.</leg:OBLIGATION>
```

Fragment 045.008.001 is annotated as an obligation incumbent (bearer="le_EC") on the legal entity le_EC (the European Commission). It is also linked to paragraphs 045.003 and 045.005 (rel="045.003 045.005"), which indicate in which cases the European Commission decides that an adequate level of protection is or Is no longer guaranteed.

PROHIBITION

Definition

A PROHIBITION is a fragment that has the semantic value of imposing a prohibition on an actor: it expresses a negative legal constraint imposed on this actor. Typical formulations are: “X must not…” or “X is prohibited from…”.

Syntax

<leg: PROHIBITION IDENTIFIER = "*identifier*" bearer = "(*identifier*) +" rel = "(*identifier*) +" except = "(*identifier*) +" is_list_header = "*boolean*" has_list_header = "*identifier*"**>*Fragment mixed content*</leg: PROHIBITION>**

As with all fragments, the content of the PROHIBITION element is mixed. The Summary table of fragment attributes lists the attributes of the element PROHIBITION which has only 2 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the bearer role introduces the identifier of the entity on which the prohibition applies (or identifiers if the role is fulfilled by several entities).

The fragment can also be related to other fragments or textual elements (attributes rel and except, see Dependency relationship and Exceptional relationship).

When the fragment enters a list, it may have an additional structural attribute (is_list_header or has_list_header, see List relationship).

Annotation recommendations

- The annotator has to analyze the semantic value of the fragment, bearing in mind that not all prohibitions contain an explicit marker.
- Distinction between PROHIBITION and PERMISSION : as prohibition is the negation of a permission, a fragment such as "X has permission to... only if… " should be annotated as prohibition – not permission – with an exception, *i.e.* in the same way as a fragment such as "It it prohibited that X … unless…" (see example of fragment 053.004.001 below).
- Identifying the bearer of a prohibition is not always straightforward: in case of doubt, it is better to note it as UNKNOWN without taking the risk of a questionable interpretation.

Examples

```
<leg:PROHIBITION IDENTIFIER="028.002.001" bearer="p_PRO">The
processor shall not engage another processor without prior specific
or general written authorisation of the controller.</leg:PROHIBITION>
```

Fragment 028.002.001 is annotated as a prohibition incumbent (bearer="p_PRO") on the person p_PRO (the processor), meaning that the processor must not do what the content of the PROHIBITION element indicates.

```
<leg:PROHIBITION IDENTIFIER="053.004.001" bearer="UNKNOWN">A member
shall be dismissed only <leg:EXCEPT>in cases of serious misconduct or
if the member no longer fulfils the conditions required for the
performance of the duties</leg:EXCEPT>.</leg:PROHIBITION>
```

In the case of fragment 053.004.001, the target is unknown (bearer="UNKNOWN") because it is not explicit in the body of the regulation, which does not specify who has the responsibility to remove, if applicable, a member of a supervisory authority (p_MSA) from his or her duties.

PERMISSION

Definition

A PERMISSION is a fragment which has the semantic value of granting permission to an actor: it expresses the fact that there is no legal constraint imposed on this actor on a particular subject. Typical formulations are: "X may ..." or "X has the right to ...".

Syntax

```
<leg: PERMISSION IDENTIFIER = "identifier" bearer = "(identifier) +"
rel = "(identifier) +" except = "(identifier) +" is_list_header =
```

"*boolean*" has_list_header = "*identifier*">***Fragment mixed content***</leg:PERMISSION>

As with all fragments, the content of the element PERMISSION is mixed. The Summary table of fragment attributes lists the attributes of the element PERMISSION which has only 2 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the bearer role introduces the identifier of the entity concerned by the permission (or the identifiers if the role is filled by several entities).

The fragment can also be related to other fragments or textual elements (attributes rel and except, see Dependency relationship and Exceptional relationship).

When the fragment enters a list, it may have an additional structural attribute (is_list_header or has_list_header, see List relationship).

Annotation recommendations

- In principle, permissions relate to persons and not to legal entities, which rather have powers (POWER).
- Distinction between permissions (PERMISSION) and exceptions (EXCEPTION): permissions can often also be considered as exceptions to obligations but the annotation as PERMISSION should be preferred if the fragment has a clear semantic value, given that the exception relation can be encoded as an except attribute of the PERMISSION fragment.
- Distinction between rights (RIGHT) and permissions (PERMISSION): both types of fragments can be expressed by the same formulations “X may ...” or “X has the right to ...” but a right differs from a permission because it results in an obligation for a third party who may be required to enforce X's right (see RIGHT).

Examples

```
<leg:PERMISSION IDENTIFIER="012.003.002" bearer="p_CONT"
except="012.003.001">That period may be extended by two further
months where necessary, taking into account the complexity and number
of the requests.</leg:PERMISSION>
```

Fragment 012.003.002 indicates that it is possible to postpone a deadline. It does not specify who has permission to do so. The fact that it is the controller (bearer="p_CONT") is known from the preceding fragment.

RIGHT

Definition

A RIGHT is a fragment which has the semantic value of conferring a right on an actor X so that X can turn to another actor Y to have his right respected. Typical formulations are: “X has the right to ...”, “X is entitled to ...” or “X can exercise the right to ... with respect to Y”.

Syntax

```
<leg: RIGHT IDENTIFIER = "identifier" bearer = "(identifier) +" target
= "(identifier) +" rel = "(identifier) +" except = "(identifier) +"
is_list_header = "boolean" has_list_header = "identifier”>Fragment
mixed content</leg: RIGHT>
```

As with all fragments, the content of the `RIGHT` element is mixed.

The Summary table of fragment attributes lists the attributes of the element `RIGHT` which has 3 mandatory attributes:

- the identifier (`IDENTIFIER`) uniquely identifies the annotated fragment;
- the role `bearer` introduces the identifier of the entity which holds the right (or the identifiers if the role is fulfilled by several entities);
- the role `target` introduces the identifier of the entity that has an obligation if the holder decides to exercise his right (or the identifiers if the role is fulfilled by several entities).

The fragment can also be related to other fragments or textual elements (attributes `rel` and `except`, see The dependency relation and The exception relation).

When the fragment enters a list, it may have an additional structural attribute (`is_list_header` or `has_list_header`, see List relationship).

Annotation recommendations

- The target of the right, the person who is obliged to respect it or make it respected, is not always explicitly mentioned in the `RIGHT` fragments. The attribute is mandatory but it may be associated with the `UNKNOWN` entity.
- Distinction between rights (`RIGHT`) and permissions (`PERMISSION`): to differentiate the two types of fragments, it is important to check that the right does translate into an obligation for the target ,even if the target may not be identified.

Examples

```
<leg:RIGHT IDENTIFIER="016.000.001" bearer="p_DS" target="p_CONT">The data subject shall have the right to obtain from the controller, without undue delay the rectification of inaccurate personal data concerning him or her.</leg:RIGHT>
```

Fragment `016.000.001` is annotated as a right of the data subject (`bearer="p_DS"`), which implementation is the responsibility of the controller (`target="p_CONT"`).

POWER

Definitions

A power is "the ability vested in a [legal entity], to use the proper means to exercise [its] [legal] competence"[7], *i.e.* its ability to take legal decisions.

There are two types of powers in the GDPR:

- A regulatory power is the ability for an actor to modify certain legal rules, in particular to amend them, strengthen them, or specify the conditions of their application. Typical formulations are: "X may specify …", "X may introduce additional conditions…";
- An executive power is the ability for an actor to take binding decisions on concrete cases. The formulations are varied.

Syntax

```
<leg: POWER IDENTIFIER = "identifier" type =(ruling|execution) bearer = "(identifier) +"  rel = "(identifier) +" except = "(identifier) +" is_list_header = "boolean" has_list_header = "identifier">fragment happy mixed</leg: POWER>
```

As with all fragments, the content of the `POWER` element is mixed. The Summary table of fragment attributes lists the attributes of the `POWER` element which has 3 mandatory attributes:

- the identifier (`IDENTIFIER`) uniquely identifies the annotated fragment;
- the `type` attribute specifies the nature of the power the actor has: `ruling` for regulatory powers and `execution` for executive powers;
- the `bearer` role introduces the identifier of the entity that holds the power to rule or decide (or the identifiers if the role is fulfilled by several entities).

[7] https://www.dictionnaire-juridique.com/definition/pouvoir.php

The fragment can also be related to other fragments or textual elements (attributes `rel` and `except`, see Dependency relationship and Exceptional relationship).

When the fragment enters a list, it may have an additional structural attribute (`is_list_header` or `has_list_header`, see List relationship).

Annotation recommendations

- The holder (`bearer`) of a power is necessarily a legal entity.
- The categories of powers (whether executive or regulatory) are to be considered as broad categories allowing to annotate both the expression of powers and everything related to the modalities of these powers or to their limitations.
- Distinction between obligations (`OBLIGATION`) or permissions (`PERMISSION`) relating to a legal entity, and powers (`POWER`): powers are involved whenever the legal entity either takes a decision that has a legal effect (`type="execution"`) or issues a legal rule (`type="ruling"`) as in Fragment `041.005.001` below: the power held by the legal entity legitimizes its future decisions or regulatory changes. If the fragment is about a legal entity but not about a legal ruling, it should be annotated as `OBLIGATION`, `PERMISSION`, `PROHIBITION`, etc. (see for example Fragment `041.004.002`, below).

Examples

```
<leg:POWER IDENTIFIER="009.004.001" type="ruling"
bearer="le_MS">Member States may maintain or introduce further
conditions, including limitations, with regard to the processing of
genetic data, biometric data or data concerning health.</leg:POWER>
```

Fragment `009.004.001` indicates that the legal entity `le_MS` (the Member States) has the power to amend the regulation (`bearer="le_MS"`). This is a regulatory power (`type="ruling"`).

```
<leg:POWER IDENTIFIER="041.005.001" type="execution"
bearer="le_SA">The competent supervisory authority shall revoke the
accreditation of a body as referred to in paragraph 1 if the
conditions for accreditation are not, or are no longer, met or where
actions taken by the body infringe this Regulation.</leg:POWER>
```

Fragment `041.005.001` indicates that the legal entity `le_SA` (the supervisory authority) has the power to revoke an approval (`bearer="le_SA"`), which has legal force. This is an executive power (`type="execution"`).

```
<leg:POWER IDENTIFIER="045.005.003" type="execution" bearer="le_EC"
except="045.005.002">On duly justified imperative grounds of urgency,
the Commission shall adopt immediately applicable implementing acts
```

```
in accordance with the procedure referred to in Article
93(3).</leg:POWER>
```

Fragment 045.005.003 indicates that the European Commission (legal entity le_EC) has an executive power ("adopts implementing acts"), which consists of repealing, amending or suspending a previous decision stating that a country or organization provides an adequate level of protection. This fragment ("immediately applicable") constitutes an exception to the previous power 045.005.002, which must be "applied in accordance with the procedure referred to in article 93".

```
<leg:OBLIGATION IDENTIFIER="035.004.002" bearer="le_SA">The
supervisory authority shall communicate those lists to the Board
referred to in Article 68.</leg:OBLIGATION>
```

```
<leg:OBLIGATION IDENTIFIER="041.004.002" bearer="le_BOD">It shall
inform the competent supervisory authority of such actions and the
reasons for taking them.</leg:OBLIGATION>
```

Fragments 035.004.002 and 041.004.002 are annotated not as powers but as obligations of the "body" (bearer="le_BOD") because the actions of "communicating" or "informing" are not legal decisions[8].

ATTRIBUTION

Definitions

A ATTRIBUTION is a fragment that has the semantic value of conferring a particular property on an actor, typically a skill, responsibility or qualification.

Syntax

<leg:ATTRIBUTION IDENTIFIER="*identifier*" type="(responsability|competency|qualification)" bearer="(*identifier*)+" rel="(*identifier*)+" except="(*identifier*)+" is_list_header="*boolean*" has_list_header="*identifier*"**>*Fragment mixed content*</leg:ATTRIBUTION>**

As with all fragments, the content of the ATTRIBUTION element is mixed. The Summary table of fragment attributes lists the attributes of the element ATTRIBUTION which has 3 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;

[8] "A decision is the act by which a legal person expresses in which direction he or she intends to act or in which direction the persons over whom he or she has authority should act" (https://www.dictionnaire-juridique.com/definition/decision.php).

- the `type` attribute specifies the nature of the property conferred on the actor (three values are possible: `responsability`, `competency`, `quality`);
- the `bearer` role introduces the identifier of the actor to whom the property is conferred (or the identifiers if the role is completed by several entities).

The fragment can also be related to other fragments or textual elements (`rel` and `except` attributes, see Dependency relationship and Exceptional relationship).

When the fragment enters a list, it may have an additional structural attribute (`is_list_header` or `has_list_header`, see List relationship).

Annotation recommendations

- Provisions which attribute particular property to an actor are annotated as `ATTRIBUTION`. They are neither deontic nor subordinate (procedure or other) provisions. Properties are attributed either to generic actors or to particular instances of actors.
- Property attributions which are neither competences nor responsibilities are annotated as quality (`type="quality"`).
- Distinction between competence-type property attributions (`ATTRIBUTION type="competency"`) and powers (`POWER`): the two notions of competency and power are close but power is stronger and should be prefered is applicable. In particular, a power may indicate not only who has the competency in a situation or a particular domain (the who) but also how this competency is exercised in practice (the what or the how), as in Example `056.003.002` below.
- Strong responsibilities ("is responsible for", see Fragment `028.004.002`) and weak responsibilities ("is held responsible for", see Fragment `082.004.001`) are annotated in the same way.

Examples

```
<leg:ATTRIBUTION IDENTIFIER="028.004.002" type="responsability"
bearer="p_PRO">Where that other processor fails to fulfil its data
protection obligations, the initial processor shall remain fully
liable to the controller for the performance of that other
processor's obligations.</leg:ATTRIBUTION>
```

```
<leg:ATTRIBUTION IDENTIFIER="082.004.001" type="responsability"
bearer="p_CONT p_PRO">Where more than one controller or processor, or
both a controller and a processor, are involved in the same
processing and where they are, under paragraphs 2 and 3, responsible
for any damage caused by processing, each controller or processor
```

```
shall be held liable for the entire damage in order to ensure
effective compensation of the  data subject.</leg:ATTRIBUTION>
```

Fragments 028.004.002 and 082.004.001 both confer a responsibility (type="responsability") on the processor and the controller (bearer = "p_PRO" and bearer="p_CONT p_PRO"). They are annotated similarly even though the former expresses a stronger responsibility than the latter.

```
<leg:ATTRIBUTION IDENTIFIER="028.010.001" type="quality"
bearer="p_PRO">Without prejudice to Articles 82, 83 and 84, if a
processor infringes this Regulation by determining the purposes and
means of processing, the processor shall be considered to be a
controller in respect of that processing.</leg:ATTRIBUTION>
```

Fragment 028.010.001 indicates when the processor (bearer="p_PRO") is requalified (type="quality") as "controller". There are few examples of the quality type, but (re-)qualification is indeed a quality attributed to a person (and not an obligation or a permission). This is neither a competence nor a responsibility in the legal sense of the term. It is a requalification, in the sense that an actor changes roles ("the processor is considered to be a 'controller '").

```
<leg:POWER IDENTIFIER="056.003.002" bearer="le_LSA"
type="execution">Within a period of three weeks after being informed
the lead supervisory authority shall decide whether or not it will
handle the case in accordance with the procedure provided in Article
60, taking into account whether or not there is an establishment of
the controller or processor in the Member State of which the
supervisory authority informed it.</leg:ATTRIBUTION>
```

Fragment 056.003.002 indicates that it is for the "lead supervisory authority" to decide how to handle a case. The fragment indicates who can decide without specifying how this power should or can be exercised. It is therefore an execution power (POWER of the "execution" type) and not a competency statement.

```
<leg:ATTRIBUTION IDENTIFIER="055.003.001" type="competency"
bearer="le_SA">Supervisory authorities shall not be competent to
supervise processing operations of courts acting in their judicial
capacity.</leg:ATTRIBUTION>
```

Fragment 055.003.001 specifies the area of competence (type="competency") of the supervisory authorities (bearer="le_SA").

DEFINITION

Definitions

A DEFINITION is a fragment that describes the specific meaning to be given to a term (simple or compound) in the text of the regulation.

Syntax

<leg:DEFINITION IDENTIFIER="*identifier*" obj="(*identifier*)+">*Fragment mixed content*</leg:DEFINITION>

As with all fragments, the content of the DEFINITION element is mixed. The Summary table of fragment attributes lists the attributes of the element DEFINITION which includes only 2 mandatory attributes:

- The identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the obj role gives the identifier of the term which is defined (possibly a list of identifiers in the case where several definitions are nested).

Annotation recommendations

Definitions of concepts or terms are often grouped in a dedicated section or chapter, such as article 4 of the GDPR (ARTICLE IDENTIFIER="004") but definitional fragments can also appear in the rest of the enacting terms.

Examples

```
<leg:DEFINITION IDENTIFIER="026.001.001" obj="p_C">Where two or more controllers jointly determine the purposes and means of processing, they shall be joint controllers.</leg:DEFINITION>
```

Fragment 026.001.001 provides the definition of the term “joint controller” (obj="p_JC").

Subordinate fragments

Subordinate fragments are fragments that can only be interpreted with reference to another fragment or textual element (fragment, paragraph, article, chapter, etc.) on which they depend. There are two main types of subordinate fragments: complements (COMPLEMENT) and exceptions (EXCEPTION).

COMPLEMENT

Definitions

A complement (COMPLEMENT) is a subordinate fragment that complements the fragment or textual element on which it depends. There are different types of such complements: procedures, text specifications but also indications of validity or impact:

- A procedural-type complement is a subordinate fragment that specifies a procedure, *i.e.* the formalities necessary for the validity of an act, decision or action that is described in the textual element on which the annotated fragment depends;
- A complement of the text specification type is a subordinate fragment that describes or specifies the content of a legal text that is mentioned in the fragment or textual element on which it depends and that contains additional provisions to the regulation being annotated;
- A precision-type complement is a subordinate fragment which provides details of the fragment on which it depends, of its content or the way it should be understood.
- A validity indication is a subordinate fragment which provides information on the validity of the provision on which it depends (date of entry in force, duration, date of revocation, etc.).
- An impact indication is a subordinate fragment which specifies the impact that the provision on which it depends has (or has not) on another provision. .

Syntax

```
<leg:COMPLEMENT IDENTIFIER="identifier"
type=(procedure|text-specification|validity|impact)
rel="(identifier)+" except="(identifier)+" is_list_header="boolean"
has_list_header="identifier">Fragment mixed content</leg:COMPLEMENT>
```

As with all fragments, the content of the LEGAL_PRECISION element is mixed. The Summary table of fragment attributes lists the attributes of the element COMPLEMENT which includes 3 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the type attribute characterizes the type of precision (four values are possible: procedure, text-specification, validity, impact),
- the rel attribute introduces the identifier of the textual element (fragment, paragraph, article, etc.) to which the legal precision relates, or a list of identifiers if the relation concerns several textual elements (see Dependency relationship).

A complement can also introduce an exception with respect to one or more other textual elements (fragment, paragraph, article, etc.), in which case an `except` attribute is added and it is associated with the identifiers of these elements (see Exceptional relationship).

When the fragment is part of a list, it may have an additional structural attribute (`is_list_header` or `has_list_header`, see List relationship).

Annotation recommendations

- Complements of the `procedure` type concern the rules, modalities or formal conditions of implementation of contracts, decisions, legal acts, obligations, permissions, etc. Procedures have a legal value and failure to comply with a procedure associated with a decision or an act may lead to the invalidation of that decision or act.

- The act or decision to which a complement of the type `text_specification` relates is not always mentioned explicitly in the textual element to which the complement relates.

- Only those fragments which specify the content of texts with legal value should be annotated as complements of the `text_specification` type. Fragment `035.007.001` below describes what a data protection impact assessment should contain, but the fragment is annotated as an obligation and not as a text specification because the text that is specified is not a legal act.

- The semantics of the `impact` type of complements is to establish a relation between two dispositions, the one to which the complement is attached and another disposition – which may be only vaguely evoked – on which the first has an impact or has no impact. In practice, complements of the `impact` type often have a negative value ("without prejudice to…").

- Distinction between the impact-type of complements (`COMPLEMENT type="impact"`) and exceptiosn (`EXCEPTION`): Any exception can be analysed as an impact, but the reverse is not true and the rule of exhaustive annotation of exceptions must prevail; the annotator must annotate as impacts only those fragments that cannot be annotated as exceptions, just as only those fragments that have no deontic value of their own are marked as complement.

```
<leg:OBLIGATION IDENTIFIER="035.007.001" bearer="p_CONT"
rel="035.001.001"><P>The assessment shall contain at least:</P>
     <LIST TYPE="alpha">
     <ITEM>
          a)a systematic description of the envisaged processing
operations and the purposes of the processing, including, where
applicable, the legitimate interest pursued by the controller;
     </ITEM>
     <ITEM>
          b)an assessment of the necessity and proportionality of
the processing operations in relation to the purposes;
```

```
      </ITEM>
      ...
      </LIST>
</leg:OBLIGATION>
```

Examples (text_specification)

```
<leg:POWER IDENTIFIER="045.003.001" type=”execution”
bearer="le_EC">The Commission, after assessing the adequacy of the
level of protection, may decide, by means of implementing act, that a
third country, a territory or one or more specified sectors within a
third country, or an international organisation ensures an adequate
level of protection within the meaning of paragraph 2 of this
Article.</leg:POWER>
```

```
<leg:COMPLEMENT IDENTIFIER="045.003.002" type="text_specification"
rel="045.003.001"> The implementing act shall provide for a mechanism
for a periodic review, at least every four years, which shall take
into account all relevant developments in the third country or
international organisation.</leg:COMPLEMENT>
```

Fragment `045.003.002` is a text specification that complements Fragment `045.003.001` in the sense that it explains what should contain the execution act referenced in that fragment.

Examples (procedure)

```
<leg:COMPLEMENT IDENTIFIER="028.009.001" type="procedure"
rel="028.003 028.004">The contract or the other legal act referred to
in paragraphs 3 and 4 shall be in writing, including in electronic
form.</leg:COMPLEMENT>
```

Fragment `028.009.001` is complement to the a text specification that specifies the text (contract or legal act) referred to previously, in two paragraphs that are referenced in the `rel` attribute (`028.003` and `028.004`). It explains how the mentioned texts should be referred to (`type="procedure".`

```
<leg:OBLIGATION IDENTIFIER="012.001.001" bearer="p_CONT">The
controller shall take appropriate measures to provide any information
referred to in Articles 13 and 14 and any communication under
Articles 15 to 22 and 34 relating to processing to the data subject
in a concise, transparent, intelligible and easily accessible form,
using clear and plain language, in particular for any information
addressed specifically to a child.</leg:OBLIGATION>
```

```
<leg:COMPLEMENT IDENTIFIER="012.001.002" type="procedure"
rel="012.001.001">The information shall be provided in writing, or by
```

```
other means, including, where appropriate, by electronic
means.</leg:COMPLEMENT>
```

Fragment `012.001.002` is a procedural complement that indicates how the information transmission obligation given by Fragment `012.001.001` is to be performed.

```
<leg:COMPLEMENT IDENTIFIER="072.001.001" type="procedure"
rel="UNDEFINED">The Board shall take decisions by a simple majority
of its members, unless otherwise provided for in this
Regulation.</leg:COMPLEMENT>
```

Fragment `072.001.001` is a procedural clarification of how decisions of the committee are to be taken. As these decision making processes are not described in the text of the GDPR, there is no fragment, paragraph or article to which to attach the procedure: the fragment is annotated as a subordinate fragment but the `rel` attribute is set to `UNDEFINED`.

Examples (validity)

```
<leg:COMPLEMENT IDENTIFIER="045.009.001" type="validity" rel="045.003
045.005">Decisions adopted by the Commission on the basis of Article
25(6) of Directive 95/46/EC shall remain in force until amended,
replaced or repealed by a Commission Decision adopted in accordance
with paragraph 3 or 5 of this Article.</leg:COMPLEMENT>
```

Fragment `045.009.001` provides a complement concerning the tiling of validity of the decisions under Directive 95/46/EC and Paragraphs `045.003` and `045.005`.

Examples (impact)

```
<leg:COMPLEMENT IDENTIFIER="045.007.001" type="impact"
rel="045.005">A decision pursuant to paragraph 5 of this Article is
without prejudice to transfers of personal data to the third country,
a territory or one or more specified sectors within that third
country, or the international organisation in question pursuant to
Articles 46 to 49.</leg:COMPLEMENT>
```

Fragment `045.007.001` explains that the decisions referred in Paragraphs `045.005` have no impact ("without prejudice") on the authorisations given in virtue of articles 46 to 49.

Examples (precision)

```
<leg:COMPLEMENT IDENTIFIER="006.003.001" type="precision"
rel="006.001.001"><P>The basis for the processing referred to in
point (c) and (e) of paragraph 1 shall be laid down by:</P>
          <LIST TYPE="alpha">
          <ITEM>
```

```
                a) Union law; or </ITEM>
            <ITEM>
                b) Member State law to which the controller is
subject.  </ITEM>
            </LIST>
            </leg:COMPLEMENT>
```

Fragment `006.003.001` is annotated in `COMPLEMENT` because it specifies the "processing referred to in paragraph 1" (`rel="006.001.001"`) without having any independent legal value. The type is `precision` because the precision is neither a procedure nor a text specification nor an impact or validity indication. Note that the granularity chosen for the annotation does not allow for a more precise reference to "points c) and e)" of Fragment `006.001.001` (see Semantic annotation and List relationship).

```
<PARAG IDENTIFIER="024.001">

            <NO.PARAG>1.</NO.PARAG> <leg:OBLIGATION
IDENTIFIER="024.001.001" bearer="p_CONT">Taking into account the
nature, scope, context and purposes of processing as well as the
risks of varying likelihood and severity for the rights and freedoms
of natural persons, the controller shall implement appropriate
technical and organisational measures to ensure and to be able to
demonstrate that processing is performed in accordance with this
Regulation.</leg:OBLIGATION> <leg:OBLIGATION IDENTIFIER="024.001.002"
bearer="p_CONT">Those measures shall be reviewed and updated where
necessary.</leg:OBLIGATION> </PARAG>

            <PARAG IDENTIFIER="024.002">

            <NO.PARAG>2.</NO.PARAG> <leg:COMPLEMENT
IDENTIFIER="024.002.001" type="precision" rel="024.001.001">Where
proportionate in relation to processing activities, the measures
referred to in paragraph 1 shall include the implementation of
appropriate data protection policies by the
controller.</leg:COMPLEMENT> </PARAG>

            <PARAG IDENTIFIER="024.003">

            <NO.PARAG>3.</NO.PARAG> <leg:COMPLEMENT
IDENTIFIER="024.003.001" type="precision" rel="024.001.001">Adherence
to approved codes of conduct as referred to in Article 40 or approved
certification mechanisms as referred to in Article 42 may be used as
an element by which to demonstrate compliance with the obligations of
the controller.</leg:COMPLEMENT> </PARAG>
```

Fragments 024.002.001 and 024.003.001 are annotated as COMPLEMENT of type precision subordinated to Fragment 024.001.001 insofar as the details provided by those fragments are not procedural in nature (which excludes the procedure type), do not relate to the content of a legal act (which excludes the text_specification type) nor to the impact or validity of Fragment 024.001.001.

EXCEPTION

Definitions

An exception (EXCEPTION) is a subordinate fragment whose semantic value can only be understood in relation to the provision on which it depends and to which it provides an exception.

Syntax

<leg:EXCEPTION IDENTIFIER="*identifier*" except="(*identifier*)+" is_list_header="*boolean*" has_list_header="*identifier*"**>*Fragment mixed content*</leg:EXCEPTION>**

As with all fragments, the content of the EXCEPTION element is mixed. The Summary table of fragment attributes lists the attributes of the element EXCEPTION which includes only 2 mandatory attributes:

- the identifier (IDENTIFIER) uniquely identifies the annotated fragment;
- the except attribute introduces the identifier of the textual element (fragment, paragraph, article, etc.) to which the exception relates, or a list of identifiers in the case where the exception relates to several textual elements (see Exceptional relationship).

When the fragment is part of a list, it may have an additional structural attribute (is_list_header or has_list_header, see List relationship).

Annotation recommendations

- EXCEPTION fragments are one of the 3 ways to annotate an exception relation, see Exceptional relationship.
- Formulations like "provisions XXX do not apply" are characteristic of EXCEPTION fragments, which have no semantic value other than introducing an exception.

Examples

```
<leg:EXCEPTION IDENTIFIER="013.004.001" except="013.001 013.002
013.003">Paragraphs 1, 2 and 3 shall not apply where and insofar as
the data subject already has the information.</leg:EXCEPTION>
```

Fragment `013.004.001` is annotated as `EXCEPTION` because its only semantic value is to introduce an exception; it depends on three paragraphs which are cited in the text (`except="013.001 013.002 013.003"`)

```
<leg:ATTRIBUTION IDENTIFIER="055.002.001" type="competency"
bearer="le_SA">Where processing is carried out by public authorities
or private bodies acting on the basis of point (c) or (e) of Article
6(1), the supervisory authority of the Member State concerned shall
be competent.</leg:ATTRIBUTION> <leg:EXCEPTION
IDENTIFIER="055.002.002" except="056">In such cases Article 56 does
not apply.</leg:EXCEPTION>
```

Fragment `055.002.002` which is reduced to the mention of an exception is annotated as `EXCEPTION`. This example is particular insofar as the article to which the exception is made appears in the GDPR after the mention of the exception (`except="056"`).

Sub-fragments

A sub-fragment is an element that identifies a portion of text within an existing fragment. Its semantics is defined in relation to the encompassing fragment and the relation is simply marked by the embedding of the sub-fragment in the encompassing fragment.

A rich annotation could multiply the number of sub-fragments to be annotated. In this guide, the emphasis is on exceptions at the expense of other types of sub-fragments, such as conditions, impact or procedural elements, which might be accomodated in a finer-grained annotation.

EXCEPT

Definition

An EXCEPT element is a sub-fragment which does not form a sentence and cannot be annotated as a fragment but which introduces an exception to the fragment in which it appears.

Syntax

The pattern below shows the except subfragment in the context of its enclosing fragment.

<leg:*FRAGMENT_ELEMENT_NAME list_of_attributes*>*Fragment Mixed content***
<leg:EXCEPT>*Fragment Mixed content* </leg:EXCEPT>***Fragment mixed content***</leg:*FRAGMENT_ELEMENT_NAME*>**

The content of the EXCEPT element is mixed. The EXCEPT element does not take any attribute. The relationship of the sub-fragment to the enclosing fragment is only marked by the embedding of XML elements.[9]

Annotation recommendations

- The annotation of the sub-fragments is deliberately limited to phrases that explicitly mark exceptions, excluding conditions or reservations which do not allow to determine precisely in which cases the general provision does not apply.
- The annotation of the sub-fragments remains optional: as they are included in the fragment expressing the general provision, the reader cannot ignore them.
- Although the phrase "without prejudice to" can sometimes be interpreted as introducing a sort of reverse exception relationship, it is agreed not to annotate the corresponding sub-fragments at this stage.
- The EXCEPT element is clearly distinguished from the EXCEPTION element (see EXCEPTION) by the fact that it is not a stand-alone fragment.

Examples

```
<leg:OBLIGATION IDENTIFIER="012.003.004" bearer="p_CONT">Where the
data subject makes the request by electronic form means, the
information shall be provided by electronic means where possible,
<leg:EXCEPT>unless otherwise requested by the data
subject</leg:EXCEPT>.</leg:OBLIGATION>
```

The subordinate clause introduced by "unless…" introduces an exception to the general rule expressed in Fragment 012.003.004, but the reservation "where possible" is not strictly speaking an exception.

[9] An exception to this rule occurs when the sub-segment EXCEPT contains a list and the xml breakdown imposes to separate the list header and the items in two different sub-segments, both typed as EXCEPT and linked by the attributes is_except_list_header="true" and is_except_list_items="true".

Relationships between fragments

General considerations

The fragments are often linked together by semantic relationships. Depending on the case, a semantic relation marks a subordination between a subordinate fragment and the fragment on which it depends or a simple relation between two semantically autonomous fragments.

Relationships between fragments are generally fragment-to-fragment relationships, but they extend to relationships between a fragment and one or more other textual elements (articles, paragraphs, chapters, sections).

In the following, a distinction is made between the *source* element of the relationship which carries the relational attribute and the *target* element towards which the relational attribute points.

The annotation introduces 2 types of semantic relationships. T

- Exception relations allow a source fragment (called exceptional) to be attached to a target element expressing a general rule whose content is replaced by that of the exceptional fragment in cases where the exception applies. Exception relations can appear in fragments of various types (*e.g.* `OBLIGATION`, `PROHIBITION`): they are mandatory for fragments of type `EXCEPTION`; they remain implicit in the fragments of type `EXCEPT` (see Exceptional relationship).

- Semantic dependency relations indicate that the source fragment (called dependent) must be interpreted in relation to the target fragment (see Dependency relationship). They can appear in any type of fragment but they are mandatory for `COMPLEMENT` fragments which are subordinate fragments and which must be linked to the fragment to which they are subordinated.

In addition to the above, list relations have no semantic value but allow annotating lists while preserving compatibility with structural markers (see List relationship).

Relations between fragments are always marked asymmetrically: only one of the two fragments refers to the other. They are annotated in the form of an attribute associated with the source fragment. This attribute takes as value the identifier(s) of the target fragment(s) or textual element(s) on which it depends.

Fragments are actually caught in a complex interplay of relationships (the regulation can be seen as a semantic web of provisions). These relationships are important for the reader to take into account, as he or she must always interpret the fragments in their context. However, as it is impossible to mark all relationships, this guide recommends that only those relationships that are not immediately accessible to the reader should be made explicit in the form of annotations. In particular, relationships between

two fragments located in close proximity to each other or relationships made explicit by a cross-reference are considered to be accessible to the reader without the need for annotation.

The different types of relationship between fragments can be combined.

Types of relationships

Dependency relationship

The `rel` dependency relationship links two fragments, one of which specifies the meaning or is interpreted in the light of the other:

- it is mandatory when it marks a subordination relation between fragments, *i.e.* for the fragments of type `COMPLEMENT`;
- it is optional when it marks a relation of semantic dependence of a non-subordonate fragment (autonomous fragment), but it is recommended when the relation is deemed important by the annotator without being immediately accessible to the reader (relationships between fragments in the immediate neighborhood of each other or relationships that are the subject of an explicit reference are considered immediately accessible to the reader)

The relationship is often expressed in the text of the dependent fragment as an anaphora which refers to all or part of the fragment or the textual element on which it depends.

Syntax (`rel` attribute)

`rel="(`*`identifier`*`)+"`

The dependency relation is marked as a `rel` attribute associated with the dependent fragment and having as value the identifier(s) of the fragment(s) or textual element(s) on which it depends.

The `rel` attribute is

- obligatory for the fragments of type `COMPELMENT`,
- prohibited for definitions (`DEFINITION`) and rights (`RIGHT`) which are independent fragments by definition,
- optional in other types of fragments.

Annotation recommendations

Except for the complements for which the `rel` attribute is mandatory, the annotation of dependency relations which are immediately accessible to the reader should be avoided, either because the

fragments appear in the same zone or “reading context”, or because the relationship is made explicit in the form of a cross-reference (*e.g.* “the objectives referred to in Article 23, paragraph 1”).

By convention, the reading context of a fragment is the paragraph in which it appears, plus the preceding paragraph if the fragment appears at the head of the paragraph, see example below.

When a non-mandatory dependency relationship is introduced, it often links the annotated fragment to a higher-order textual element (a paragraph or an article) rather than to a simple fragment.

Examples of reading context

```
<PARAG IDENTIFIER="012.001">
      <NO.PARAG>1.</NO.PARAG>
      <leg:OBLIGATION IDENTIFIER="012.001.001" ...>
      ...
      </leg:OBLIGATION>
      <leg:COMPLEMENT IDENTIFIER="012.001.002" ... rel="012.001.001">
      ...
      </leg:COMPLEMENT>
      <leg:PERMISSION IDENTIFIER="012.001.003" ...
except="012.001.002">
      ...
      </leg:PERMISSION>
</PARAG>

<PARAG IDENTIFIER="012.002">
      <NO.PARAG>2.</NO.PARAG>
      <leg:OBLIGATION IDENTIFIER="012.002.001" ...>
      ...
      </leg:OBLIGATION>
      <leg:PROHIBITION IDENTIFIER="012.002.002" ...>
      ...
      </leg:PROHIBITION>
 </PARAG>
```

The reading context of Fragment `012.002.002` is the paragraph `012.002` in which it appears, but the reading context of Fragment `012.002.001` is the sequence of the two paragraphs `012.001` and `012.002` to enable the reader to relate Fragment `012.002.001` to those who precede it. Among relationships local to the reading context, only those that are mandatory need to be annotated.

Examples of dependency relationship annotations

```
<leg:OBLIGATION IDENTIFIER="045.008.001" bearer="le_EC" rel="045.003
045.005">The Commission shall publish in the Official Journal of the
European Union and on its website a list of the third countries,
territories and specified sectors within a third country and
international organisations for which it has decided that an adequate
level of protection is or is no longer ensured.</leg:OBLIGATION>
```

The OBLIGATION fragment 045.008.001 is annotated as semantically dependent on the two paragraphs 045.003 and 045.005 which are remote (outside the reading context of Fragment 045.008.001). The text of the fragment does not contain any marker of this relationship but the understanding of the fragment allows to clarify and annotate the relation: the two paragraphs define the powers of the Commission in deciding whether a general level of adequate protection is ensured.

```
<leg:COMPLEMENT IDENTIFIER="012.001.002" type="procedure"
rel="012.001.001">The information shall be provided in writing, or by
other means, including, where appropriate, by electronic
means.</leg:COMPLEMENT>
```

Fragment 012.001.002 is annotated as subordinate to the immediately preceding Fragment 012.001.001 because the rel attribute is mandatory for COMPLEMENT elements. Dependency is marked in the text by the anaphoric repetition of "provide any information" (from 012.001.001) in "Information shall be provided"

```
<leg:COMPLEMENT IDENTIFIER="072.001.001" type="procedure"
rel="UNDEFINED">The Board shall take decisions by a simple majority
of its members, unless otherwise provided for in this
Regulation.</leg:COMPLEMENT>
```

As Fragment 072.001.001 is a procedure (COMPLEMENT of type procedure), it relates to another fragment and the subordination relationship must be marked (the rel attribute is mandatory). However, as the fragment does not indicate and does not allow inferring which fragments describe the decisions the committee takes, the relationship points to an undefined reference (rel="UNDEFINED").

Exceptional relationship

Whenever a fragment introduces an exception to one or more fragment(s) or textual element(s), the exception relationship must be marked, but it can take different forms (see Annotation recommendations below).

Like all relationships, the exception one is asymmetrical: the fragment introducing the exception explicitly refers to the textual element(s) to which it makes an exception, but the reverse link is not marked.

Syntax (except attribute)

except="(*identifier*)+"

The except attribute marks an exception relationship between the fragment with which it is associated and the fragment(s) or textual element(s) whose identifiers it takes as values.

The except attribute is

- mandatory for `EXCEPTION` fragments which are subordinate fragments;
- optional in other types of fragments.

Annotation recommendations

- Even if the exception relations are sometimes difficult to spot, their chaining should be made as explicit as possible in the annotation to allow the readers to make the link between two provisions where one introduces an exception to the other.
- There are 3 ways to annotate an exception relationship and to mark the fact that a fragment or sub-fragment B introduces an exception to a provision expressed in a textual element A:
    - if Element B is a semantically autonomous fragment (`OBLIGATION`, `PERMISSION` type fragment, etc.) with its own semantic value of obligation, permission, etc., it should be annotated according to its type (`OBLIGATION`, `PERMISSION`, etc.) and linked to A by an `except` attribute (see Syntax (except attribute)) ;
    - if Element B mainly expresses opposition to Element A without having any semantic value of its own, it should be annotated as a subordinate fragment of type `EXCEPTION` linked to A by an `except` attribute (see EXCEPTION);
    - If Element B is a portion of A and therefore does not constitute an independent sentence, it should be annotated as a sub-fragment, *i.e*. an `EXCEPT` element with no attribute inserted in the content of Element A (see EXCEPT).

Examples

```
<leg:EXCEPTION IDENTIFIER="006.001.002" except="006.001.001">Point
(f) of the first subparagraph shall not apply to processing carried
out by public authorities in the performance of their
tasks.</leg:EXCEPTION>
```

Fragment `006.001.002` introduces an explicit exception (“point f) of the first paragraph does not apply …”). As the mentioned “point f)” is not a referable fragment, the exception is annotated as covering the entire “first paragraph”, *i.e.* Fagment `006.001.001`.

```
<leg:COMPLEMENT type="validity" IDENTIFIER="092.005.001" rel="012.008
043.008">A delegated act adopted pursuant to Article 12(8) and
Article 43(8) shall enter into force only <leg:EXCEPT>if no objection
has been expressed by either the European Parliament or the Council
within a period of three months of notification of that act to the
European Parliament and the Council or if, before the expiry of that
period, the European Parliament and the Council have both informed
```

```
the Commission that they will not
object</leg:EXCEPT>.</leg:COMPLEMENT>

<leg:COMPLEMENT type="validity" IDENTIFIER="092.005.002" rel="012.008
043.008" except="092.005.001">That period shall be extended by three
months at the initiative of the European Parliament or of the
Council.</leg:COMPLEMENT>
```

In this example, the second fragment, a `COMPLEMENT` of type `validity` (`092.005.002`) introduces an exception to the first (fragment `092.005.001`). The exception relationship is marked in the text by the anaphoric phrase "This period is extended by three months ..." which refers to the "period of three months of notification ..." mentioned in the first fragment. Note that the first validity complement has two exceptions: one is marked by a subfragment, the `EXCEPT` element contained in Fragment `092.005.001`, and the other is marked by the `except` attribute associated with Fragment `092.005.002` (`except="092.005.001"`).

List relationships

Whenever possible, a list of items and its introductory sentence are contained in the same fragment. A list relation is used to reconstitute a list in its uni, *i.e.* to link the two parts when they belong to distinct XML elements. This relationship has a purely syntactic role : it is imposed by the XML structuring of the source document which follows the formal division into paragraphs.

Since list relationships are introduced during the pre-annotation phase, the annotator usually does not have to deal with them. He or she simply needs to keep them unchanged and be aware of the semantic unity of the linked fragments. Ah exception, however, is the case where the annotator detects an `EXCEPT` subfragment that spans the introductory phrase and the item list (see the technical manual for details of the procedure)

Syntax (list header and item list attributes)

`is_list_header="true"`[10]

`has_list_header="identifier"`

`is_except_list_header="true"`

`is_except_list_items="true"`

[10] The `is_list_header`, `is_except_list_header` and `is_except_list_items` attributes can actually take 2 values (false or true) but false value is by default so the attribute is only used in its positive form (`is_list_header="true"`).

When a list spans over multiple XML elements and cannot be annotated as a stand-alone XML element, the introductory phrase and the item list are annotated as separate fragments,of the same semantic value, and the relation is marked by associating

- an `is_list_header="true"` attribute to the introductory element that introduces the items;
- a `has_list_header` attribute to the item list with a reference ot the introductory element as value;

The two other attributes `is_except_list_header="true"` and `is_except_list_items="true"` have a similar role for encoding the lists within `EXCEPT` subfragments.

Two cases arise :

- If there is no EXCEPT, the items constitute a block annotated as a single fragment linked by the `has_list_header` attribute to the introductory fragment (`is_list_header="true"`);.
- TIf there is an EXCEPT, the items constitute a block annotated as a separate subfragment (`has_except_list_items="true"`) of a separate fragment (with has_list_header attribute) and implicitly linked to the introductory subfragment (`has_except_list_header="true"`) of the encompassing header fragment (with `is_list_header="true"`).

Examples

```
<P><leg:OBLIGATION IDENTIFIER="028.003.001" bearer="p_CONT
p_PRO">Processing by a processor shall be governed by a contract or
other legal act under Union or Member State law, that is binding on
the processor with regard to the controller and that sets out the
subject-matter and duration of the processing, the nature and purpose
of the processing, the type of personal data and categories of data
subjects and the obligations and rights of the
<controller.</leg:OBLIGATION> <leg:COMPLEMENT
IDENTIFIER="028.003.002" type="text_specification"
is_list_header="true" rel="028.003.001">That contract or other legal
act shall stipulate, in particular, that the
processor:</leg:COMPLEMENT></P>
<leg:COMPLEMENT IDENTIFIER="028.003.003" type="text_specification"
has_list_header="028.003.002" rel="028.003.001">
     <LIST TYPE="alpha">
     <ITEM>
          a)processes the personal data only on documented
instructions from the controller, [...] ;       </ITEM>
```

```
      <ITEM>
            b)ensures that persons authorised to process the personal
data have committed themselves to confidentiality or are under an
appropriate statutory obligation of confidentiality;
      </ITEM>
      [...]
      </LIST>
</leg:COMPLEMENT>
```

In this example, the list of items is annotated as a single fragment `028.003.003` which is of type `COMPLEMENT` like the introductory fragment `028.003.002`. However, as the two parts are not part of the same `P` element, the entire list cannot be annotated as a single `COMPLEMENT` element and it is split in two elements.

Annotation of entities

General perspective

The provisions of the GDPR mention many entities, mainly concepts, persons and legal entities.

The goal is not to annotate the mentions of entities within the text of the GDPR but to relate the entities to the fragments in which they play a key role: the concepts that are the subject of a definition or the actors that are directly involved in the provisions, *i.e.* that are thus expected to appear as an attribute value in the fragments which are annotated.

Each entity is represented by one and only one entry (of a specific type) in a dictionary. This entry introduces a unique identifier, which is referenced whenever that entity plays a role in a fragment.

The annotation of entities involves 2 different tasks:

- the maintenance of dictionaries which introduces entity identifiers in the form of dictionary entries (see Dictionary): the annotator is provided with starting dictionaries for concepts and actors but new entries can be added as necessary
- the specification of the role that the entities play in the fragments where they are involved (see Roles or fragment to entity relationships).

There are 3 main types of entities in the GDPR: concepts (`CONCEPT`) and two kinds of actors: persons (`PERSON`) and legal entities (`LEGAL_ENTITY`):

- a **concept** is a notion which is the subject of a definition and thus plays a particular role in the regulation;

- a **person** entity is a person who plays a particular role in the regulation; it is usually a generic person, whether individual ("the data subject", "the subcontractor") or collective ("the subcontractors");
- a **legal entity** is an entity which plays a particular legal role in the regulation but which is not a person; it is usually an institution or body and it can be a specific entity ("the European Parliament") or a generic entity ("the supervisory authority").

In addition to those, there are also abstract entities, whose type is undefined and which are introduced for reference purposes.

All entities have a unique identifier (`id="prefix_identifier"`), with a prefix corresponding to the type of entity. The abstract entities have special identifiers (*e.g.* `UNKNOWN`) which are used as role values.

Components of entity annotation

Dictionary

The dictionary is an XML element without textual content that is added to the GDPR to list all the entities introduced in the annotation of the regulation and to associate them with unique identifiers. In practice and for convenience, the dictionary is implemented in two external files: `actorsDictionary.xml` and `conceptsDictionary.xml`.

Dictionary entries

There are two categories of entries:

- entries that introduce the concrete entities of the concept (`CONCEPT_ENTRY`), person (`PERSON_ENTRY`) and legal entity (`LEGAL_ENTITY_ENTRY`) types, for which as many identifiers as necessary can be introduced and to which linguistic labels can be associated;
- entries that introduce abstract entities (`ABSTRACT_ENTITY_ENTRY`) used for reference purposes in fragment roles and for which only a small number of special identifiers, such as `UNKNOWN` and `ALL`, should be defined.

Syntax

There are 4 types of entity entries:

```
<leg:ABSTRACT_ENTITY_ENTRY
id="(UNKNOWN|ALL|abstract_entity_identifier)"/>
```

```
<leg:CONCEPT_ENTRY id="c_identifier">
     (<LABEL lang="language" value="text"/>)*
</leg:CONCEPT_ENTRY>


<leg:PERSON_ENTRY id="p_identifier">
     (<LABEL lang="language" value="text"/>)*
</leg:PERSON_ENTRY>


<leg:LEGAL_ENTITY_ENTRY id="le_identifier">
     (<LABEL lang="language" value="text"/>)*
</leg:LEGAL_ENTITY_ENTRY>
```

Dictionary entries (`..._ENTRY`) are elements without textual content. They are empty elements, except for the entries of concrete entities which can contain one or more sub-elements (`LABEL`) associating the entities with the linguistic forms they can take in different languages.

The prefix of the identifiers of concrete entities corresponds to the type of the entity to which they are associated: `c_`, `p_` and `le_` respectively for the identifiers of concepts, persons and legal entities.

Annotation recommendations

- When an annotator discovers in the text of the regulation either an entity that he or she considers as important but which is not yet in the dictionary, he or she should update the dictionary accordingly.
- It is recommended to associate at least one label to each concrete entity entry.

  Even if the label associated with an entity is often the canonical linguistic form of this entity, there can be several relevant labels (*e.g.* variation of number, abbreviation, anaphoric repetition) for an entity even for the same language.

Exemple

```
<leg:DICTIONARY >
     <leg:ABSTRACT_ENTITY_ENTRY id="UNKNOWN"/>
     <leg:PERSON_ENTRY id="p_DS">
          <LABEL lang="FR" value="la Personne Concernée"/>
          <LABEL lang="EN" value="The data subject"/>
     </leg:PERSON_ENTRY>
     <leg:PERSON_ENTRY id="p_CONT"/>
     <leg:LEGAL_ENTITY_ENTRY id="le_EP"/>
</leg:DICTIONARY>
```

Roles or fragment to entity relationships

Definition

Roles are the relationships that entities play in provisions or more precisely in fragments.

Syntax and signature

The roles that entities play in provisions are marked by relationships linking fragments to entities. They are expressed in the form of attributes which are associated with fragments and which take an entity identifier as value.

Each type of fragment has a particular signature, *i.e.* a set of possible roles, some of which are mandatory and others are optional. This signature is specified in the Summary table of fragment attributes.

Role semantics

- the **object** role (`obj`) introduces what is the object of the annotated fragment, in particular the concept defined in the case of a `DEFINITION` fragment;
- the **bearer** role (`bearer`) introduces the actor primarily concerned by a provision, *e.g*. the one who is granted a permission (fragment of type `PERMISSION`), who holds a right (fragment of type `RIGHT`) or who has an obligation (fragment of type `OBLIGATION`).
- the **target** role (`target`) introduces an actor concerned on a secondary basis by a provision, next to the main actor: this role is used in particular to specify whom responsibility it is to enforce a right as the right of a person (attribute `bearer` of a fragment of type `RIGHT`) results in an obligation for another person (target attribute).

Annotation recommendations

- When an entity plays a role in a fragment, there is usually a mention of that entity in the text of the fragment. The annotation then consists of
    - checking that an entry corresponding to that entity is in the dictionary with a specific identifier;
    - associating this entity identifier with the fragment as the value of the corresponding role.
- When a fragment has a mandatory role but the entity that performs this role is not clearly identified, an abstract entity (*e.g.* `UNKNOWN` of type `ABSTRACT_ENTITY_ENTRY`) is used as the value of this role.

- The value of a role attribute can be a unique identifier or a list of identifiers if multiple entities play the same role in the fragment.

Entity types

Several types of entities are distinguished in the annotation:

- concrete entities: concepts, person and legal entities;
- abstract entities.

CONCEPT

Definition

A **concept** is an entity corresponding to a notion which plays a particular role in the regulation and may also be the subject of a definition.

Each concept is associated with a unique identifier which is introduced in the dictionary and used as attribute value in the fragments where the concept plays a role.

Annotation recommendations

- All concept identifiers must be entered in the dictionary.
- All definitions relate to an entity but only those that are neither persons nor legal entities are annotated as `CONCEPT`.
- As indicated in Dictionary and Mentions of entities, not all the concepts mentioned in the regulation are intended to be the subject of an entry in the dictionary. In practice, it is recommended at least to associate an entry in the dictionary with the concepts which are the object of a definition.

Examples

```
<leg:CONCEPT_ENTRY id="c_PPD"><LABEL lang="FR" value="Traitement de données à caractère personnel"/><LABEL lang="EN" value="Processing of Personal Data"/></leg:CONCEPT_ENTRY>
```

PERSON

Definition

A **person** entity refers to a person who plays a particular role in the regulation. This is generally a generic person, either individual or collective.

Each person-type entity is associated with a unique identifier which is introduced in the dictionary and used as attribute value in the fragments where the person plays a role.

Annotation recommendations

- All the identifiers of persons must be entered in the dictionary.
- As indicated in Dictionary and Mentions of entities, not all the persons mentioned in the regulation are intended to be the subject of an entry in the dictionary. In practice, it is recommended at least to associate an entry in the dictionary with the persons who play a key role in certain fragments (`DEFINITION`, `OBLIGATION`, etc.) and which identifiers are thus to be used as value of fragment attributes.

Examples

```
<leg:PERSON_ENTRY id="p_DS"><LABEL lang="FR" value="la personne
concernée"/><LABEL lang="EN" value="the Data
Subject"/></leg:PERSON_ENTRY>
```

```
<leg:RIGHT IDENTIFIER="016.000.002" bearer="p_DS"
target="p_CONT">Taking into account the purposes of the processing,
the data subject shall have the right to have incomplete personal
data completed, including by means of providing a supplementary
statement.</leg:RIGHT>
```

LEGAL_ENTITY

Definition

A legal entity is an entity which plays a particular legal role in the regulation and is not considered as a person. It is usually an institution or a body. It can be a specific entity (“European Parliament”) or a generic entity (“supervisory authority”).

Annotation recommendations

- All the identifiers of legal entities must be entered in the dictionary.

- As indicated in Dictionary and Mentions of entities, not all the legal entities mentioned in the regulation are intended to be the subject of an entry in the dictionary. In practice, it is recommended at least to associate entries in the dictionary with the legal entities which play a key role in certain fragments (`DEFINITION`, `POWER`, etc.) and which identifiers are thus to be used as value of fragment attributes.

- Not all mentions of legal entites necessarily have the same form. As legal entities have a well-defined status, there is little variation but abbreviations or abbreviated anaphoric forms can be found, especially when the full name of the entity is long and the context is unambiguous, as in the following examples:
    - After the first mention, which is complete, the “European Commission” is systematically referred to as the “Commission” in the GDPR.
    - The “European Data Protection Board” is referred to as the “Board”.

Examples

```
<leg:LEGAL_ENTITY_ENTRY id="le_EU"><LABEL lang="FR" value="Union
Européenne"/><LABEL lang="EN" value="European
Union"/></leg:LEGAL_ENTITY_ENTRY>
```

```
<leg:OBLIGATION IDENTIFIER="006.003.004" bearer="le_EU le_MS">The
Union or the Member State law shall meet an objective of public
interest and be proportionate to the legitimate aim
pursued.</leg:OBLIGATION>
```

Abstract entity (UNKNOWN and ALL)

The annotation uses abstract entity whenever it is necessary to refer to an entity (as the value of a mandatory role for example) without being able to identify it precisely.

These abstract entities have their own identifier:

- `UNKNOWN` is used as a value for attributes representing entity roles (*e.g.* `obj`, `bearer`, `target`) when the entity is not clearly identified.
- ALL is used as a value for attributes representing entity roles (*e.g.* `obj`, `bearer`, `target`) referring to an undefined group of entities.
-

Examples

```
<leg:ABSTRACT_ENTITY_ENTRY id="UNKNOWN"/>
```

```
<leg:RIGHT IDENTIFIER="022.001.001" bearer="p_PC"
target="UNKNOWN">The data subject shall have the right not to be
subject to a decision based solely on automated processing, including
```

```
profiling, which produces legal effects concerning him or her or
similarly significantly affects him or her.</leg:RIGHT>
```

In Fragment `022.001.001` of type `RIGHT`, the target role is mandatory but the fragment does not indicate who should ensure (`target = "UNKNOWN"`) that the right of the person concerned (`bearer="PC"`) is respected.

Attributes (Synopsis)

The attributes and relational roles are presented above (see Section Relationships between fragments. and Section Roles or fragment to entity relationships.) but there are some other attributes :

- **Identifiers** allow to associate a unique key with an element :
    - The `IDENTIFIER` attribute is used to identify a fragment or another textual element (paragraph or article);
    - The `id` attribute is used to identify entities.
- **The** `type` **attribute** specifies the semantic value of certain fragments and takes its value in a set of chosen keywords:
    - `competency`, `responsability` or `quality` for `ATTRIBUTION` fragments;
    - `ruling` or `execution` for `POWER` fragments;
    - `procedure`, `text-specification`, `validity` or `impact` for `COMPLEMENT` fragments.
- A**ttributes marking the List relationship** :
    - The `is_list_header`, `is_except_list_header` attributes are Boolean attributes set to false (”`false`”) by default; their value is set to `true` to indicate that the associated fragment or `EXCEPT` subfragment introduces a list;
    - The `has_list_header` attribute takes a fragment identifier as its value. It is associated to a fragment that makes up a list and links it to the introductory fragment of the list;
    - The `is_except_item_list` attribute is a Boolean attribute set to false by default; its value is set to `true` to indicate that the associated `EXCEPT` subfragment forms a list of items that must be linked to the introductory subfragment.

Summary table of fragment attributes

	obj	target	bearer	type	except	rel	IDENTIFIER	is_list_header	has_list_header
DEFINITION	r						r	o	o
RIGHT		r	r				r	o	o
OBLIGATION			r		o	o	r	o	o
PERMISSION			r		o	o	r	o	o
PROHIBITION			r		o	o	r	o	o
POWER			r	r	o	o	r	o	o
QUALITY_ATTRIBUTION			r	r	o	o	r	o	o
COMPLEMENT				r	o	r	r	o	o
EXCEPTION					r		r	o	o

	is_except_list_header	is_except_item_list
EXCEPT (subfragment)	o	o

r=required, o=optional

Appendices

XML in short

The XML annotation of a text consists of inserting start and end markers or "tags" surrounding the text that must be annotated. Those *tags* together with the enclosed text form an *element*[11]. In addition to its name, the start tag can be associated with attributes.

The *content* of an XML element refers to text and/or sub-elements that are surrounded by its start and end tags. An alternate notation allows merging start and end tags for items with empty content[12].

An XML element can be entirely contained within another or be separate from it, but overlapping is prohibited[13]. An xml document form a tree (*i.e.* all elements are descendants of a single one, called the root). The child elements of an XML element areits direct sub-elements

[11] `<XXX>xxx xxx</XXX>` where `XXX` is the name of the element, and "`xxx xxx`" is the annotated text.
[12] `<XXX/>` is equivalent to `<XXX></XXX>`.
[13] One can have `<A>... <B>...</B>... </A>` or `<A>... </A>... <B>...</B>` but not `<A>... <B>...</A>...</B>`.

To standardize annotations and reduce errors, a *schema* (described in an `.xsd` file) may specify the available tags and attributes, as well as legal inclusions between elements, along with constraints on the order of child elements. The schema can reuse one or more standard vocabularies (preexisting schemas identified by a standard URI). To avoid name collisions between vocabularies, a *namespace* is generally associated with each vocabulary: all the names of a given vocabulary receive the same prefix (ending with ":") which abbreviates a standard URI.

Interested readers can refer to XML tutorlal for additional and mode technical information.

XML schemas (XSD files)

XML Schemas are used to describe and validate the structure and the content of XML data. An XML schema defines the elements, attributes and data types that are used in XML annotation (annotation vocabulary) and explain how they can be used (annotation rules). By validating an XML file with its associated schema, one can also control that the XML annotations conform to the annotation language defined in the schema.

The semantic definitions and rules designed for encoding legal sources and described in the present guide are formally encoded in a specific XML schema: the `GDPR_SemanticSchema.xsd` file.

The description of that schema is beyond the scope of this guide but the interested reader can refer to the World Wide Web Consortium (W3C) XML schema tutorial and open the `GDPR_SemanticSchema.xsd` file to see how it formalises the semantic annotation language presented in this guide.

The technical organization and the links between the annotated GDPR, the semantic schema and the various associated files is described in Organization of XML and XSD files.

GDPR structural annotation

The text of the GDPR is published in an XML format whose schema[14] provides the structuring vocabulary. This is a fairly comprehensive schema, used for many documents and only a part of which is used in the GDPR. For the ease of annotation, we provide a simplified version.

The root element is `ACT`, which contains the definitions, the title and the operative part of the regulation (`ENACTING_TERMS`) on which this guide focuses.

The basic overall XML structure of simplified GDPR is the following:

```
<ACT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://formex.publications.europa.eu/schema/forme
x LightS&S_MergedSchema.xsd" some namespaces>
```

[14] http://formex.publications.europa.eu/schema/formex-05.55-20141201.xd

```
      <TITLE> ... </TITLE>
      <ENACTING.TERMS> ... </ENACTING.TERMS>
      <FINAL> ... </FINAL>
</ACT>
```

The text of the enacting terms (ENACTING_TERMS) is highly structured. High-level structures are marked by a generalist element (DIVISION), which is used for both chapters and sections, some chapters themselves being subdivided into sections. Technically, a division is made up of a title (with an indication of its level) and a sequence of divisions or articles.

An article (ARTICLE) consists of a title, an optional sub-title and either a sequence of paragraphs that contain text, or directly text[15]. A paragraph (PARAG) is numbered while text has no numbering. On the other side, the text is possibly formatted by elements (P and LIST), the schema allowing cascaded inclusions.

ARTICLE and PARAG elements have a unique IDENTIFIER attribute formed as a combination of article and paragraph numbers and used for reference. On the contrary the superstructures (divisions corresponding to the chapters and sections) and the sub-structures (P and LIST) of the paragraphs have no identifier: precisely referring to them is not possible.

Example

```
<ARTICLE IDENTIFIER="001">
  <TI.ART>Article 1</TI.ART>
  <STI.ART>Subject-matter and objectives</STI.ART>
  <PARAG IDENTIFIER="001.001">
    <NO.PARAG>1.</NO.PARAG>
     This Regulation lays down rules relating to the protection of
natural persons with regard to the processing of personal data and
rules relating to the free movement of personal data.
  </PARAG>
  <PARAG IDENTIFIER="001.002">
    <NO.PARAG>2.</NO.PARAG>
      This Regulation protects fundamental rights and freedoms of
natural persons and in particular their right to the protection of
personal data.
  </PARAG>
```

This example shows the beginning of Article 1 in the simplified xml version of the text with its structure annotations. The article has a title (“Article 1”), a sub-title (“Subject-matter and objectives”) and three numbered paragraphs (the first two are reproduced here).

[15] The SUBDIV and COMMENT elements, which are allowed in the content of an ARTICLE element by the schema, do not appear in the GDPR.

Organization of XML and XSD files

Semantic annotations are controlled by a schema organised in several XSD files. The annotators do not need to open or modify those schema files but the interested reader can nevertheless have a look at them.

This section describes the overall organisation of the XML and XSD files for the semantic annotation of the GDPR.

There are 3 XML files :

- The main file is `Light_GDPR_EN_xxx.xml` (or `Light_GDPR_EN_xxx_working.xml` for the working version). It is both structurally and semantically annotated. The annotator has to enrich it with semantic annotations without modifying the preexisting structural annotations.
- The previous file is associated with dictionaries that are included automatically:
    - The `ActorDictionary.xml` file contains the identifiers of the actors (`PERSON` and `LEGAL_ENTITY`).
    - The `ConceptDictionary.xml` file contains the identifiers of the concepts.

Those XML files are associated with five schema files:

- The main schema is `GDPR_SemanticSchema.xsd`: it gives the formal definition of the semantic vocabulary and rules described in the present semantic annotation guide.
- Only one schema is directly referenced in the main XML file: `LightS&S_MergedSchema.xsd`. It describes the light structural annotation language and dynamically integrates the semantic language with it by importing the following schema.
- The `LightS_SemanticSchemaWrapper.xsd` schema is a technical wrapper of `GDPR_SemanticSchema.xsd`: it imports it and makes it compatible with the light structural annotation language.
- There are two working versions of the previous schemas (`LightS&S_MergedSchema_working.xsd` and `LightS_SemanticSchemaWrapper_working.xsd`) that enable the validation of in progress annotation (with remaining neutral FRAGMENT annotations inherited from pre-annotation).

The semantic vocabulary and rules presented in the present guide are therefore formalised in the `GDPR_SemanticSchema.xsd` schema, which is automatically included in the `Working_LightS&S_MergedSchema.xsd` schema, which also includes the description of structural annotations.

To integrate the semantic vocabulary and the structural vocabulary, namespace are used: the structure vocabulary is not prefixed, while the semantic annotations are prefixed by `leg:`[16], which makes it possible to distinguish the two vocabularies without further inspection.

To read, validate, complete or modify the annotations of the GDPR, this set of files must be installed in the working directory.

[16] The only element name that does not need to be prefixed is that of the `label` element that appears in some entries of the dictionary.