GDPR / RGPD Semantic Annotation Guide

François Lévy, Adeline Nazarenko

Prerequisites

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 subject, related to a particular actor or related to another provision. It consists in segmenting the provisions into 1 or more fragments of text, associating a type to those fragments, identifying the key entities which play a role in them and the inter-fragment relationships which are not given simply by the ordering of the sentences and paragraphs.

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 are therefore obvious to the reader. .

General principles

The semantic annotation consists in associating elements of a semantic vocabulary to pieces of text.

Semantic vocabulary

The semantic annotation attaches three categories of information to the text:

- It marks up the **fragments** : these 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 in an autonomous way.
    - 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.
- It also identifies the **entities** that play an important role in the regulation: their references are annotated in the text of the regulation, particularly in the provisions where they play a key role.
- It makes explicit two types of **relationships** as attributes of certain fragments:
    - Inter-fragment **relations** – notably exception relations or precision relations that link a procedure or modality to the provision that it specifies – which generalize to relations between fragments and larger textual elements (paragraphs, articles, etc.);
    - **roles** that link a fragment to an entity, to indicate who has an obligation, who has a right or power, etc.

Here is the list of the semantic vocabulary elements to be used in semantic annotation. Each element is described in detail in the rest of this guide with a focus on how to use it for annotation. A description including the elements having a technical role is given in the appendix of this guide (The semantic vocabulary).

Fragments	Autonomous fragments	OBLIGATION
		PROHIBITION
		PERMISSION
		RIGHT
		POWER
		QUALITY_ATTRIBUTION
		DEFINITION
	Subordinate fragments	LEGAL_PRECISION

		EXCEPTION
	Sub-fragments	EXCEPT
Entities	Actors	PERSON
		LEGAL_ENTITY
	Concepts	CONCEPT
Relations	Inter-fragment relations	rel
		except
		is_list_header
		has_list_header
	Roles	obj
		bearer
		target

Annotation coverage

All the articles of the regulation are 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 are annotated: this includes but is not limited to those that play a role in a fragment.
- 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 relations of exceptions and those that relate to legal precision.

Adding semantic annotations

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

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` which is worth `v2`). 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. 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 and those specific to RGPD annotation system are presented in the following, in the form of patterns as shown in the next section.

Practical details

The annotations are added in the file `GDPR_EN_S&S_annotation.xml`, where the text structure annotations already appear. The annotator is not expected to modify the existing annotations (they are part of the reference text) but to add a finer-grained semantic annotation layer[1].

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).

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 also subdivided into `ALINEA` (subparagraphs), `TEXT` and `P` elements. The semantic annotations are mainly to be inserted in the text of the paragraphs and must respect some constraints. The constraints related to the form of the semantic elements are indicated in their description by a pattern built on the model of the generic pattern below:

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

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

</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. More detailed explanations about patterns can be found in the appendix (Patterns describing the schema of semantic annotations).

There are two main types or semantic annotation elements : the fragments and sub-fragments, the entities and the related dictionary of entities:

- A fragment can only be marked in the content of a structural element `ALINEA`, `TEXT` or `P` and a sub-fragment in the content of a fragment.
- An entity mention can only be marked in a fragment or a sub-fragment or in a structural element of type `P`, `STI.ART` or `VISA`. All the mentions of a given entity refer to its unique identifier that is given in the entity dictionary. For convenience of annotation, the dictionary entries are kept in separate files which are linked to the main XML document and two dictionaries are provided, respectively for the actors and the concepts (`actorsDictionary.xml` and `conceptsDictionary.xml`). The annotator should keep the dictionary files open and progressively add the entities discovered during the annotation..

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[2] and each fragment is associated with a semantic type. As indicated in the prerequisites, this type 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.

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

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.

In addition, a special identifier (UNDEFINED) is predefined to allow reference to undefined fragments or textual elements (see Structure of the provided GDPR file).

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[3]

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

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

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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).

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

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 (but 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 often appears as an entity mention in the text, 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 <leg:PERSON
ref="p_CONT">controller</leg:PERSON> is able to demonstrate that it
is not in a position to identify the <leg:PERSON ref="p_PC">data
subject</leg:PERSON>, the controller shall inform the <leg:PERSON
ref="p_PC">data subject</leg:PERSON> 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
<leg:PERSON ref="p_PC">data subject</leg:PERSON> makes the request by
electronic form means, the information shall be provided by
electronic means where possible, <leg:EXCEPT>unless otherwise
requested by the <leg:PERSON ref="p_PC">data
subject</leg:PERSON></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 <leg:LEGAL_ENTITY
ref="le_EC">Commission</leg:LEGAL_ENTITY> shall publish in the <HT
TYPE="ITALIC">Official Journal of the European Union</HT> 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) and 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 text between the start and end tags can include the following XML elements:

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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 (see example of fragment `053.004.001` below).
- identifying the bearer of a prohibition is not always straightforward: if in doubt it is best to note it as `UNKNOWN` without taking the risk of a questionable interpretation.

Examples

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

Fragment 028.002.001 is annotated as a prohibition incumbent (bearer = "p_PRO) on 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
<leg:PERSON ref="p_MSA">member</leg:PERSON> 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 text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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 be interpreted as exceptions to obligations but the annotation as PERMISSION should be preferred if the fragment has a clear semantic value, even if it means introducing an exception relation as an except attribute.
- 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 without specifying 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 text between the start and end tags can include the following XML elements:

- elements of the original annotation linked to the structure (`LIST`), typographical marks (`HT`, `QUOT.START`, `QUOT.END`) or dates (`DATE`),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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 or a target if the latter is not identified.

Examples

```
<leg:RIGHT IDENTIFIER="016.000.001" bearer="p_PC" target="p_CONT">The
<leg:PERSON ref="p_PC">data subject</leg:PERSON> shall have the right
to obtain from the <leg:PERSON ref="p_CONT">controller</leg:PERSON>,
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_PC"`), 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”[4], *i.e*. its ability to take legal decisions.

There are two types of powers in the GDPR:

- 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…”;
- executive power is the ability for an actor to take binding decisions on concrete cases. The formulations are varied.

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

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 text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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).

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 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 to 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">
<leg:LEGAL_ENTITY ref="le_MS">Member States</leg:LEGAL_ENTITY> 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 <leg:LEGAL_ENTITY
ref="le_SA">supervisory authority</leg:LEGAL_ENTITY>  shall revoke
the accreditation of a  <leg:LEGAL_ENTITY
ref="le_BOD">body</leg:LEGAL_ENTITY>  as referred to in paragraph 1
if the conditions for accreditation are not, or are no longer, met or
where actions taken by the <leg:LEGAL_ENTITY
ref="le_BOD">body</leg:LEGAL_ENTITY> 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 <leg:LEGAL_ENTITY
ref="le_EC">Commission</leg:LEGAL_ENTITY> 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
<leg:LEGAL_ENTITY ref="le_SA">supervisory
authority</leg:LEGAL_ENTITY> shall communicate those lists to the
```

```
<leg:LEGAL_ENTITY ref="le_EDPB">Board</leg:LEGAL_ENTITY> referred to
in Article 68.</leg:OBLIGATION>

<leg:OBLIGATION IDENTIFIER="041.004.002" bearer="le_BOD">It shall
inform the competent <leg:LEGAL_ENTITY ref="le_SA">supervisory
authority</leg:LEGAL_ENTITY> 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 "supervisory authority" (`bearer = "le_SA"`) and of the "body" (`bearer = "le_BOD"`) because the actions of "communicating" or "informing" are not legal decisions[5].

QUALITY_ATTRIBUTION

Définition

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

Syntaxe

<leg:QUALITY_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:RIGHT>**

As with all fragments, the content of the `QUALITY_ATTRIBUTION` element is mixed. The text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (`LIST`), typographical marks (`HT`, `QUOT.START`, `QUOT.END`) or dates (`DATE`),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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

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

[5] "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 quality conferred on the actor (three values are possible: `responsability`, `competency`, `qualification`);
- the `bearer` role introduces the identifier of the actor to whom the quality 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 qualities to an actor are annotated as `QUALITY_ATTRIBUTION`. They are neither deontic nor subordinate (procedure or other) provisions. Qualities are attributed either to generic actors or to particular instances of actors.
- Quality attributions which are neither competences nor responsibilities are annotated as qualifications (`type="qualification"`).
- Distinction between competence-type quality attributions (`QUALITY_ATTRIBUTION type="competency"`) and powers (`POWER`): the two notions of competence and power are close but power is broader because it indicates not only who has the competence in a situation or a particular domain (the who) but also how this competence is exercised in practice (the what or the how).
- 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.

Exemples

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

```
<leg:QUALITY_ATTRIBUTION IDENTIFIER="082.004.001"
type="responsability" bearer="p_CONT p_PRO">Where more than one
<leg:PERSON ref="p_CONT">controller</leg:PERSON> or <leg:PERSON
```

```
ref="p_PROs">processor</leg:PERSON>, or both a <leg:PERSON
ref="p_CONT">controller</leg:PERSON> and a <leg:PERSON
ref="p_PRO">processor</leg:PERSON>, are involved in the same
processing and where they are, under paragraphs 2 and 3, responsible
for any damage caused by processing, each <leg:PERSON
ref="p_CONT">controller</leg:PERSON> or <leg:PERSON
ref="p_PROs">processor</leg:PERSON> shall be held liable for the
entire damage in order to ensure effective compensation of the
<leg:PERSON ref="p_PC">data
subject</leg:PERSON>.</leg:QUALITY_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:QUALITY_ATTRIBUTION IDENTIFIER="028.010.001"
type="qualification" bearer="p_PRO">Without prejudice to Articles 82,
83 and 84, if a <leg:PERSON ref="p_PRO">processor</leg:PERSON>
infringes this Regulation by determining the purposes and means of
processing, the processor shall be considered to be a <leg:PERSON
ref="p_CONT">controller</leg:PERSON> in respect of that
processing.</leg:QUALITY_ATTRIBUTION>
```

Fragment `028.010.001` indicates when the processor (`bearer = "p_PRO"`) is requalified (`type = "qualification"`) as “controller”. There are few examples of the qualification 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 called requalification, in the sense that an actor changes roles (“the processor is considered to be a ‘controller ’”).

```
<leg:QUALITY_ATTRIBUTION IDENTIFIER="056.003.002" bearer="le_LSA"
type="competency">Within a period of three weeks after being informed
the <leg:LEGAL_ENTITY ref="le_LSA">lead supervisory
authority</leg:LEGAL_ENTITY> 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 <leg:PERSON ref="p_CONT">controller</leg:PERSON> or <leg:PERSON
ref="p_PRO">processor</leg:PERSON> in the <leg:LEGAL_ENTITY
ref="le_MS">Member State</leg:LEGAL_ENTITY> of which the
<leg:LEGAL_ENTITY ref="le_SA">supervisory
authority</leg:LEGAL_ENTITY> informed it.</leg:QUALITY_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 a question of a competence (QUALITY_ATTRIBUTION of the "competency" type) and not of a power.

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

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

DEFINITION

Définition

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.

Syntaxe

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

As with all fragments, the content of the DEFINITION element is mixed. The text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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.

Exemples

```
<leg:DEFINITION IDENTIFIER="026.001.001" obj="p_JC">Where two or more
<leg:PERSON ref="p_CONT">controllers</leg:PERSON> jointly determine
the purposes and means of processing, they shall be <leg:PERSON
ref="p_JC">joint controllers</leg:PERSON>.</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: precisions (`LEGAL_PRECISION`) and exceptions (`EXCEPTION`).

LEGAL_PRECISION

Définition

A legal precision (`LEGAL_PRECISION`) is a subordinate fragment that complements the fragment or textual element on which it depends. There are different types of legal specifications: procedures, text specifications and other details:

- a procedural-type legal precision 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 legal precision of the text specification type is a subordinate fragment that describes or specifies the content of a legal text mentioned in the fragment or textual element on which it depends;
- other legal precisions are typed as “default”.

Syntaxe

<leg:LEGAL_PRECISION IDENTIFIER="*identifier*"
type=(procedure|text-specification|default) rel=”(*identifier*)+”

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

As with all fragments, the content of the LEGAL_PRECISION element is mixed. The text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (LIST), typographical marks (HT, QUOT.START, QUOT.END) or dates (DATE),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

The Summary table of fragment attributes lists the attributes of the element `DEFINITION` which includes 3 mandatory attributes:

- the identifier (`IDENTIFIER`) uniquely identifies the annotated fragment;
- the `type` attribute characterizes the type of precision (three values are possible: `procedure`, `text-specification`, `default`),
- 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 legal precision 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

- Legal precisions 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 legal specification of the type `text_specification` relates is not always mentioned explicitly in the textual element to which the legal precision relates.
- Only those fragments which specify the content of texts with legal value should be annotated as legal precisions 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.

```
<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>
          <NP>
               <NO.P>a)</NO.P>
               <TXT>a systematic description of the envisaged
processing operations and the purposes of the processing, including,
where applicable, the legitimate interest pursued by the <leg:PERSON
ref="p_CONT">controller</leg:PERSON>;</TXT>
          </NP>
     </ITEM>
     <ITEM>
          <NP>
               <NO.P>b)</NO.P>
               <TXT>an assessment of the necessity and
proportionality of the processing operations in relation to the
purposes;</TXT>
          </NP>
     </ITEM>
     ...
     </LIST>
</leg:OBLIGATION>
```

Exemples (text_specification)

```
<leg:LEGAL_PRECISION IDENTIFIER="028.009.001"
type="text_specification" 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:LEGAL_PRECISION>
```

Fragment `028.009.001` is 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`).

```
<leg:POWER IDENTIFIER="045.003.001" type="execution"
bearer="le_EC">The <leg:LEGAL_ENTITY
ref="le_EC">Commission</leg:LEGAL_ENTITY>, 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:LEGAL_PRECISION 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:LEGAL_PRECISION>
```

Fragment `045.003.002` is a text specification that specifies what the execution act referenced in Fragment `045.003.001` that is annotated as an execution-type power should contain.

Exemples (procedure)

```
<leg:OBLIGATION IDENTIFIER="012.001.001" bearer="p_CONT">The
<leg:PERSON ref="p_CONT">controller</leg:PERSON> 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 <leg:PERSON ref="p_PC">data
subject</leg:PERSON> 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:LEGAL_PRECISION 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:LEGAL_PRECISION>
```

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

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

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`.

Exemples (default)

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

```
            <NP>
                  <NO.P>a)</NO.P>
                  <TXT><leg:LEGAL_ENTITY
ref="le_EU">Union</leg:LEGAL_ENTITY> law; or</TXT>
            </NP>
      </ITEM>
      <ITEM>
            <NP>
                  <NO.P>b)</NO.P>
                  <TXT><leg:LEGAL_ENTITY ref="le_MS">Member
State</leg:LEGAL_ENTITY> law to which the <leg:PERSON
ref="p_CONT">controller</leg:PERSON> is subject.</TXT>
            </NP>
      </ITEM>
      </LIST>
</leg:LEGAL_PRECISION>
```

Fragment `006.003.001` is annotated in `LEGAL_PRECISION` because it specifies the "processing referred to in paragraph 1" (`rel="006.001.001"`) without having any independent legal value. The type is `default` because the precision is neither a procedure nor a text specification. 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>
      <ALINEA><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
<leg:PERSON ref="p_CONT">controller</leg:PERSON> 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></ALINEA>
</PARAG>
<PARAG IDENTIFIER="024.002">
      <NO.PARAG>2.</NO.PARAG>
      <ALINEA><leg:LEGAL_PRECISION IDENTIFIER="024.002.001"
type="default" 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 <leg:PERSON
ref="p_CONT">controller</leg:PERSON>.</leg:LEGAL_PRECISION></ALINEA>
</PARAG>
```

```
<PARAG IDENTIFIER="024.003">
      <NO.PARAG>3.</NO.PARAG>
      <ALINEA><leg:LEGAL_PRECISION IDENTIFIER="024.003.001"
type="default" 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
<leg:PERSON
ref="p_CONT">controller</leg:PERSON>.</leg:LEGAL_PRECISION></ALINEA>
</PARAG>
```

Fragments `024.002.001` and `024.003.001` are annotated in `LEGAL_PRECISION` of type `default` and as subordinate to Fragment `024.001.001` insofar as the details provided by these fragments are not procedural in nature (which excludes the `procedure` type) and do not relate to the content of a legal act (which excludes the `text_specification` type).

EXCEPTION

Définition

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.

Syntaxe

<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 text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (`LIST`), typographical marks (`HT`, `QUOT.START`, `QUOT.END`) or dates (`DATE`),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

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 Relation d'exception.
- Formulations like "provisions XXX do not apply" are characteristic of `EXCEPTION` fragments, which have no semantic value other than introducing an exception.

Exemples

```
<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 <leg:PERSON ref="p_PC">data subject</leg:PERSON> 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:QUALITY_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 <leg:LEGAL_ENTITY ref="le_SA">supervisory
authority</leg:LEGAL_ENTITY> of the <leg:LEGAL_ENTITY
ref="le_MS">Member State</leg:LEGAL_ENTITY> concerned shall be
competent.</leg:QUALITY_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 normally 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 or procedural elements, which might be accomodated in a finer-grained annotation.

EXCEPT

Definition

An EXCEPT element is a sub-fragment which is not autonomous, has no semantic value of its own, but 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 text between the start and end tag can include the following XML elements:

- elements of the original annotation linked to the structure (`LIST`), typographical marks (`HT`, `QUOT.START`, `QUOT.END`) or dates (`DATE`),
- semantic elements related to entities (see Mentions of entities) or sub-fragments (see Sub-fragments).

The `EXCEPT` element does not take an attribute. The relationship of the sub-fragment to the enclosing fragment is only marked by the embedding of XML elements.

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 an 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.

Exemples

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

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 3 types of relationships. The first two are semantic relations, the last is necessary for the annotation of certain lists:

- 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 are 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 `LEGAL_PRECISION` type fragments which are subordinate fragments and which must be linked to the fragment to which they are subordinate.

- 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 `LEGAL_PRECISION`;

- 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 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.

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

Annotation recommendations

Except for the legal precisions 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>
      <ALINEA>
      <leg:OBLIGATION IDENTIFIER="012.001.001" ...>
            ...
      </leg:OBLIGATION>
      <leg:LEGAL_PRECISION IDENTIFIER="012.001.002" ...>
            ...
     </leg:LEGAL_PRECISION>
      </ALINEA>
</PARAG>
<PARAG IDENTIFIER="012.002">
      <NO.PARAG>2.</NO.PARAG>
      <ALINEA>
```

```
      <leg:OBLIGATION IDENTIFIER="012.002.001" ...>
            ...
      </leg:OBLIGATION>
      <leg:PROHIBITION IDENTIFIER="012.002.002" ...>
            ...
      </leg:PROHIBITION>
      </ALINEA>
</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.

Examples of annotations

```
<leg:OBLIGATION IDENTIFIER="045.008.001" bearer="le_EC" rel="045.003
045.005">The <leg:LEGAL_ENTITY
ref="le_EC">Commission</leg:LEGAL_ENTITY> shall publish in the <HT
TYPE="ITALIC">Official Journal of the European Union</HT> 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 markers of this relationship. It is the interpretation of the fragment that allows the relation to be clarified and annotated: the two paragraphs define the powers of the Commission in deciding whether a general level of adequate protection is ensured.

```
<leg:LEGAL_PRECISION 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:LEGAL_PRECISION>
```

Fragment `012.001.002` is annotated as subordinate to the immediately preceding Fragment `012.001.001` because the `rel` attribute is mandatory for `LEGAL_PRECISION` 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:LEGAL_PRECISION IDENTIFIER="072.001.001" type="procedure"
rel="UNDEFINED">The <leg:LEGAL_ENTITY
ref="le_EDPB">Board</leg:LEGAL_ENTITY> shall take decisions by a
```

simple majority of its members, unless otherwise provided for in this Regulation.</leg:LEGAL_PRECISION>

As Fragment 072.001.001 is a procedure (LEGAL_PRECISION 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 hereunder).

Like all relationships, the exception relationship 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.* fFagment `006.001.001`.

```
<leg:LEGAL_PRECISION type="procedure" 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 <leg:LEGAL_ENTITY
ref="le_EP">European Parliament</leg:LEGAL_ENTITY> or the
<leg:LEGAL_ENTITY ref="le_CEU">Council</leg:LEGAL_ENTITY> within a
period of three months of notification of that act to the
<leg:LEGAL_ENTITY ref="le_EP">European Parliament</leg:LEGAL_ENTITY>
and the <leg:LEGAL_ENTITY ref="le_CEU">Council</leg:LEGAL_ENTITY> or
if, before the expiry of that period, the <leg:LEGAL_ENTITY
ref="le_EP">European Parliament</leg:LEGAL_ENTITY> and the
<leg:LEGAL_ENTITY ref="le_CEU">Council</leg:LEGAL_ENTITY> have both
informed the <leg:LEGAL_ENTITY
ref="le_EC">Commission</leg:LEGAL_ENTITY> that they will not
object</leg:EXCEPT>.</leg:LEGAL_PRECISION> <leg:LEGAL_PRECISION
type="procedure" 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 <leg:LEGAL_ENTITY ref="le_EP">European
Parliament</leg:LEGAL_ENTITY> or of the <leg:LEGAL_ENTITY
ref="le_CEU">Council</leg:LEGAL_ENTITY>.</leg:LEGAL_PRECISION>
```

In this example, the second fragment, a `LEGAL_PRECISION` of type `procedure` (`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 from the notification ..." mentioned in the first fragment. Note that the first procedure has two exceptions: one is marked by 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 relationship

The list relation is used to reconstitute a list in its unity, *i.e.* to link the introductory sentence and the elements of the list when these are contained in 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.

Syntax (is_list_header and has_list_header attributes)

`is_list_header="true"`[6]

`has_list_header="identifier"`

When a list is spread over multiple XML elements and the entire list cannot be annotated as a stand-alone XML element, the introductory sentence and the list of items are annotated as separate fragments, each according to its own 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 whose value is the identifier of the introductory element.

The annotation can take two different forms:

- the items constitute a block and are annotated as a single fragment linked to the introductory fragment by the `has_list_header` attribute;
- the items are annotated as separate fragments, each with its own semantic value and `has_list_header` attribute.

Annotation recommendations

- If the list as a whole is linked to other textual elements, this relationship is carried by the introductory element alone.
- Items elements do not accept relationships other than the list relationship to the introductory element.

---

[6] The `is_list_header` attribute can actually take 2 values of your choice (false or true) but the value false is by default so the attribute is only explained in its positive form (`is_list_header="true"`).

Examples

```
<P><leg:OBLIGATION IDENTIFIER="028.003.001" bearer="p_CONT
p_PRO">Processing by a <leg:PERSON ref="p_PRO">processor</leg:PERSON>
shall be governed by a contract or other legal act under
<leg:LEGAL_ENTITY ref="le_EU">Union</leg:LEGAL_ENTITY> or
<leg:LEGAL_ENTITY ref="le_MS">Member State</leg:LEGAL_ENTITY> law,
that is binding on the <leg:PERSON ref="p_PRO">processor</leg:PERSON>
with regard to the <leg:PERSON ref="p_CONT">controller</leg:PERSON>
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
<leg:PERSON
ref="p_CONT">controller</leg:PERSON>.</leg:OBLIGATION><leg:LEGAL_PREC
ISION 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 <leg:PERSON
ref="p_PRO">processor</leg:PERSON>:</leg:LEGAL_PRECISION></P>
<leg:LEGAL_PRECISION IDENTIFIER="028.003.003"
type="text_specification" has_list_header="028.003.002"
rel="028.003.001">
     <LIST TYPE="alpha">
     <ITEM>
          <NP>
               <NO.P>a)</NO.P>
               <TXT>processes the personal data only on documented
instructions from the <leg:ACTOR ref="p_CONT">controller</leg:ACTOR>,
[...] ;</TXT>
          </NP>
     </ITEM>
     <ITEM>
          <NP>
               <NO.P>b)</NO.P>
               <TXT>ensures that persons authorised to process the
personal data have committed themselves to confidentiality or are
under an appropriate statutory obligation of confidentiality;</TXT>
          </NP>
     </ITEM>
     [...]
     </LIST>
</leg:LEGAL_PRECISION>
```

In this first example, the list of items is annotated as a single fragment `028.003.003` which is of type `LEGAL_PRECISION` like the introductory fragment `028.003.002`. However, as the latter is part of a distinct `P` element, the entire list cannot be annotated as a single `LEGAL_PRECISION` element.

```
<P><leg:PROHIBITION
IDENTIFIER="012.005.001"bearer="p_CONT">Information provided under
Articles 13 and 14 and any communication and any actions taken under
Articles 15 to 22 and 34 shall be provided free of
charge.</leg:PROHIBITION><leg:PERMISSION IDENTIFIER="012.005.002"
is_list_header="true" bearer="p_CONT" except="012.005.001">Where
requests from a <leg:PERSON ref="p_PC">data subject</leg:PERSON> are
manifestly unfounded or excessive, in particular because of their
repetitive character, the <leg:PERSON
ref="p_CONT">controller</leg:PERSON> may either:</leg:PERMISSION></P>
<LIST TYPE="alpha">
      <ITEM>
      <NP>
            <NO.P>a)</NO.P>
            <TXT><leg:PERMISSION IDENTIFIER="012.005.003"
has_list_header="012.005.002" bearer="p_CONT">charge a reasonable fee
taking into account the administrative costs of providing the
information or communication or taking the action requested;
</leg:PERMISSION> or</TXT>
      </NP>
      </ITEM>
      <ITEM>
      <NP>
            <NO.P>b)</NO.P>
            <TXT><leg:PERMISSION IDENTIFIER="012.005.004"
has_list_header="012.005.002" bearer="p_CONT">refuse to act on the
request.</leg:PERMISSION></TXT>
      </NP>
      </ITEM>
</LIST>
```

In this second example, the two list items are annotated as separate fragments (`012.005.003` and `012.005.004`) linked by the `has_list_header` attribute to the introductory fragment `012.005.002`. They are both permissions (`PERMISSION`) like the introductory fragment `012.005.002` but they could be of different types if their semantic value required it. Since the introductory fragment is part of a separate `P` element, the entire list cannot be annotated as a single `PERMISSION` element.

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 all the entities that are mentioned in the GDPR but only those that the annotator considers the most important: in particular those that are the subject of a definition or that are directly involved in the provisions, *i.e.* that play a key role and 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 a mention of that entity is to be annotated in the text or that entity plays a role in a fragment.

The annotation of entities involves 3 different tasks:

- the maintenance of dictionaries which introduces entity identifiers in the form of dictionary entries (see Dictionary),
- the annotation of mentions of entities in the text of the provisions (see Mentions of entities),
- the clarification 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 actors who are either persons (`PERSON`) or 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`, can be defined.

Syntaxe

There are 4 types of entity entries:

```
<leg:ABSTRACT_ENTITY_ENTRY id="(UNKNOWN|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

- Even if the label associated with an entity is often the canonical linguistic form of this entity, the mentions can be varied, including in a given language (*e.g.* variation of number, abbreviation, anaphoric repetition); several labels can be introduced for an entity even for the same language.
- 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, or an unrecorded variation of an existing entity, he or she should update the dictionary accordingly.
- As indicated in the introduction of Annotation of entities, not all the entities mentioned in the document need to be annotated but at least those which are the subject of a definition or which play a key role in a fragment should be associated with a dictionary entry.

Exemple

```
<leg:DICTIONARY >
      <leg:ABSTRACT_ENTITY_ENTRY id="UNKNOWN"/>
      <leg:PERSON_ENTRY id="p_PC">
            <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>
```

Mentions of entities

The mentions of the entities whose identifier is entered in the dictionary are annotated with a reference to this identifier.

Definition

An entity mention is a segment of text that refers to a uniquely identified entity, whether particular or generic. All entities (concepts, people and legal entities) can be mentioned in texts, excluding abstract entities.

Syntaxe

<leg:ENTITY_TYPE ref="*identifier*">*Entity mixed content*</leg:ENTITY_TYPE>

Regardless of its type (`CONCEPT`, `PERSON` or `LEGAL_ENTITY`), the content of an entity mention is text with possible typographic tags (`HT`, `QUOT.START`, `QUOT.END`).

Annotation recommendations

- An entity mention is annotated as a `CONCEPT`, `PERSON` or `LEGAL_ENTITY` element which refers (attribute `ref`) to the identifier of the entity entry in the dictionary and which surrounds the portion of text mentioning the entity.
- Entity mentions may be annotated semi-automatically but care must be taken to take into account possible linguistic variations in the mentions (risk of under-annotation) and to avoid annotating homonymous expressions that do not refer to the target entity (risk of over-annotation).
- The same entity can be mentioned by different character strings but the annotator must ensure that it is the same entity.
- All the mentions of a given entity do not necessarily have to be annotated but it is recommended to annotate at least those which appear in the fragments where the entity plays an explicit role and is associated with an attribute.

Exemples

```
<leg:OBLIGATION IDENTIFIER="012.002.001" bearer="p_CONT">The
<leg:PERSON ref="p_CONT">controller</leg:PERSON> shall facilitate the
exercise of <leg:PERSON ref="p_PC">data subject</leg:PERSON> rights
under Articles 15 to 22.</leg:OBLIGATION>
```

The text of Fragment `012.002.001` indicates that the obligation lies with the “data controller”, an entity of type `PERSON` associated with the identifier `p_CONT`. The role that the entity plays is marked by the attribute (`bearer = "p_CONT"`) associated with the fragment. The entity being explicitly mentioned in the text of the fragment, this mention is itself annotated (`<leg: PERSON ref = "p_CONT">controller</leg: PERSON>`). As the other person mentioned in the fragment is not playing a particular role in the obligation, its annotation (`<leg: PERSON ref = "p_PC">data subject</ leg: PERSON>`) is not obligatory in this particular fragment.

Roles or fragment to entity relationships

Definition

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

Syntax and signature of roles

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 as value an entity identifier.

Each type of fragment has a particular signature, *i.e.* a set of possible roles, some of which are mandatory and others 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 who is obliged to respect 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;
    - annotating the mention by giving it the entity identifier as a reference (`ref` attribute);
    - associating this same 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, the abstract entity `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 to which the various mentions of the concept in the text refer as well as the fragments where it plays a role.

The `CONCEPT` element annotates the mentions of concepts in the text of the regulation and should not be confused with the `CONCEPT_ENTRY` element which introduces the identifiers of the concepts in the dictionary.

Syntax

<leg:CONCEPT ref="c_*identifier*">*Entity mixed content*</leg:CONCEPT>

The content of the `CONCEPT` element is mixed: it is the text of the annotated mention which may contain typographic tags (`HT`, `QUOT.START`, `QUOT.END`).

The `CONCEPT` element takes a single mandatory attribute `ref` referring to the identifier of the concept (**ref="c_*identifier*"**). This identifier must be prefixed by `c_` and introduced in the dictionary (see Dictionary).

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 see 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 and not all the mentions

of the concepts introduced in the dictionary are intended to be annotated. In practice, it is recommended at least to associate an entry in the dictionary with the concepts which are the object of a definition and to annotate their mentions in the fragments of the DEFINITION type where they play a key role.

Examples

Note: There is no example concept in the part of the regulation currently annotated. The only definition is for a person.

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 to which the various mentions of the entity in the text refer, as well as the fragments in which it plays a role.

The PERSON element annotates the mentions of persons in the text of the regulations and should not be confused with the PERSON_ENTRY element which introduces the identifiers of persons in the dictionary.

Syntax

<leg:PERSON ref="p_*identifier*">*Entity mixed content*</leg:PERSON>

The content of the PERSON element is mixed: it is the text of the annotated mention which may contain tags of a typographical nature (HT, QUOT.START, QUOT.END).

The PERSON element takes a single mandatory attribute ref referring to the identifier of the concept (**ref="p_*identifier*"**). This identifier must be prefixed by p_ and introduced in the dictionary (see Dictionary).

Annotation recommendations

- All the identifiers of persons must be entered in the dictionary.
- As indicated in see 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 and not all the mentions of the persons introduced in the dictionary are intended to be annotated. In practice, it is recommended at least to associate an entry in the dictionary with the people who play a key role

in certain fragments (DEFINITION, OBLIGATION, etc.) and to annotate their mentions in the fragments where they play this key role.

- The annotation of mentions of persons can be automated but not all mentions necessarily have the same form. There may in particular be variations in case, morphological variations (variations in number, for example) as in the following examples for the "controller":
  <leg:PERSON ref="p_CONT">controller</leg:PERSON>
  <leg:PERSON ref="p_CONT">Controller</leg:PERSON>
  <leg:PERSON ref="p_CONT">controllers</leg:PERSON>
  Cases of abbreviated anaphoric reference are rarer because the language of regulations is fairly regular for the mentions of key persons: for example, we do not find a case where "the data subject" is referred to as "the subject".

Exemples

```
<leg:PROHIBITION IDENTIFIER="012.002.002" bearer="p_CONT">In the
cases referred to in Article 11(2), the <leg:PERSON
ref="p_CONT">controller</leg:PERSON> shall not refuse to act on the
request of the <leg:PERSON ref="p_PC">data subject</leg:PERSON> for
exercising his or her rights under Articles 15 to 22,
<leg:EXCEPT>unless the <leg:PERSON
ref="p_CONT">controller</leg:PERSON> demonstrates that it is not in a
position to identify the <leg:PERSON ref="p_PC">data
subject</leg:PERSON></leg:EXCEPT>.</leg:PROHIBITION>
```

4 mentions of persons are annotated in Fragment 012.002.002. The p_CONT entity is mentioned twice (<leg:PERSON ref="p_CONT">controller</leg:PERSON>), as is the p_PC entity (<leg:PERSON ref="p_PC">data subject</leg:PERSON>) but the prohibition bears only on the first (bearer = "p_CONT"). The mentions of p_PC are annotated because "the data subject" plays a key role in the GDPR but its annotation in this particular fragment is not mandatory.

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").

Syntax

<leg:LEGAL_ENTITY ref="le_*identifier*">*Entity mixed content*</leg:LEGAL_ENTITY>

The content of the LEGAL_ENTITY element is mixed: it is the text of the annotated mention which may contain tags of a typographical nature (HT, QUOT.START, QUOT.END).

The LEGAL_ENTITY element takes a single mandatory attribute ref referring refers to the identifier of the concept (**ref="le_*identifier*"**). This identifier must be prefixed by le_ andintroduced in the dictionary (see Dictionary).

Annotation recommendations

- All the identifiers of legal entities must be entered in the dictionary.
- As indicated in see Dictionary and Mentions of entities, not all the entities mentioned in the regulation are intended to be the subject of an entry in the dictionary and not all the mentions of the legal entities introduced in the dictionary are intended to be annotated. 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 to annotate their mentions in the fragments where they play this key role.
- The annotation of legal entity mentions can be automated but not all mentions necessarily have the same form: as legal entities have a well-defined status, there is little variation in case or morphology but there are abbreviations or abbreviated anaphoric repetitions, 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 text.
      ```
      <leg:LEGAL_ENTITY ref="CE">European Commission</leg:LEGAL_ENTITY>
      <leg:LEGAL_ENTITY ref="CE">Commission</leg:LEGAL_ENTITY>
      ```
    - The “European Data Protection Board” is referred to as the “Board” :
      ```
      <leg:LEGAL_ENTITY ref="CEPD">European Data Protection Board</leg:LEGAL_ENTITY>
      <leg:LEGAL_ENTITY ref="CEPD">Board</leg:LEGAL_ENTITY>
      ```

Exemples

```
<leg:LEGAL_ENTITY ref="le_EP">European Parliament</leg:LEGAL_ENTITY>

<leg:LEGAL_ENTITY ref="le_SA">supervisory authority</leg:LEGAL_ENTITY>

<leg:LEGAL_ENTITY ref="le_LSA">lead supervisory authority</leg:LEGAL_ENTITY>
```

Abstract entity (UNKNOWN)

The annotation uses an UNKNOWN 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.

This abstract entity has `UNKNOWN` as its identifier. `UNKNOWN` is used as a value for attributes representing entity roles (*e.g.* `obj`, `bearer`, `target`) when the entity is not clearly identified.

An abstract entity cannot have a mention.

Examples

```
<leg:RIGHT IDENTIFIER="022.001.001" bearer="p_PC"
target="UNKNOWN">The <leg:PERSON ref="p_PC">data subject</leg:PERSON>
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.

All the attributes

There are three types of attributes:

- relational attributes marking relationships between fragments,
- roles, which link fragments to entities,
- other attributes, without semantic relation value.

Attributes relating fragments

See before, Section Relationships between fragments.

Role attributes

See before, Section Roles or fragment to entity relationships.

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.
- **Attributes marking the List relationship** :
    - the `is_list_header` attribute is a Boolean attribute set to false (“`false`”) by default; `header = ”true”` is used to mark the fact that a fragment introduces a list;
    - the `has_list_header` attribute takes a fragment identifier as its value. It is associated to a fragment that makes up or is part of a list and links it to the introductory fragment of the list.
- **The `type` attribute** specifies the semantic value of certain fragments and takes as value a chosen keyword:
    - `competency`, `responsability` or `quality` for `QUALITY_ATTRIBUTION` fragments;
    - `ruling` or `execution` for `POWER` fragments;
    - `procedure`, `text-specification` or `default` for `LEGAL_PRECISION` fragments.

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
Legal_Precision				r	o	r	r	o	o
Exception					r		r	o	o

r=required, o=optional

Appendices

Appendix 1 : Annotation project rationale

Semantic vocabulary

The vocabulary to be used for the semantic annotation includes the following elements:

- **Fragments** are sentences or pieces of sentences that make up provisions; the annotation delimits and types them, as obligations, prohibitions, powers, procedures or legal precisions. There are different sorts of fragments
    - The **autonomous fragments** have their own legal semantics and are interpreted autonomously;
    - The **subordinate fragments** have a legal value that refers to another fragment;
    - The **sub-fragments** are portions of sentences or fragments which play a particular role in the interpretation of the enclosing fragment;
- E**ntities** that play a key role in the regulation or in certain provisions should also be annotated: their mentions are annotated in the text of the regulation, in particular in the provisions in which they play a role.
- Two types of **relationships** are annotated:
    - **relations between fragments** – in particular relations of exceptions, or relations of precision introducing the procedure a provision – which generalize to relations between fragments and larger textual elements (paragraphs, articles, etc.);

- the **roles** that link a fragment to an entity to indicate upon whom an obligation is incumbent, who holds a right or power, etc.

In addition to those elements, the annotation introduces:

- a **dictionary** listing the identifiers of the entities which are referenced in the text of the regulation, either as a mention or as a role value;
- unique **identifiers** for all fragments or entities to allow them to be referenced;
- a **type** attribute that specifies the semantics of certain fragment annotations.

The tables below list all the elements and XML attributes that make up the semantic vocabulary to be used for the annotation. The guide details the definition, syntax and annotation guidelines for each of these elements and attributes.

Elements	Fragments	Autonomous Fragments	OBLIGATION
			PROHIBITION
			PERMISSION
			RIGHT
			POWER
			QUALITY_ATTRIBUTION
			DEFINITION
		Subordinate Fragments	LEGAL_PRECISION
			EXCEPTION
		Sub-fragments	EXCEPT
	Entities	Actors	PERSON
			LEGAL_ENTITY

		Concepts	CONCEPT
	Dictionary	The Dictionary element	DICTIONARY
		Entries	LEGAL_ENTITY_ENTRY
			CONCEPT_ENTRY
			ABSTRACT_ENTITY_ENTRY

Attributes	Relations	Relations between fragments	rel
			except
			is_list_header
			has_list_header
		Roles	obj
			bearer
			target
	Identifiers	Fragment Identifiers	IDENTIFIER
		Entity Identifiers	id
	Type		type

Manual vs. Automatic annotation

The present guide is intended to provide guidance to annotators in the manual annotation of the GDPR. The annotator should follow those general principles and guidelines to provide an annotation as consistent and compliant as possible.

The issue of possible automation of the annotation should not be considered at the time of manual annotation. As the resulting annotation (after arbitration in case of disagreement between annotators) is to be used as a standard to study the feasibility of an automatic annotation and to evaluate it, the manual annotation should be done without anticipating what may be easy / difficult to automate.

Use of the annotated text

The annotation is meant to ease the exploration of the regulation and the selection of provisions by the readers who are interested in law. The annotation makes it possible to select text fragments on the basis of semantic criteria in addition to keywords. The user can thus find all the obligations incumbent on X and dealing with Y (`OBLIGATION` fragments with the role `bearer="X"` and containing the keyword Y). The segmentation and structuring of the text also makes it possible to present each fragment in its context, by showing the paragraph into which it appears, possibly extended to the preceding paragraph, and by highlighting the relationships of exception or dependence into which it enters.

Annotators

The targeted annotators are not seasoned lawyers but people who have a minimal academic background and are sufficiently familiar with the legal texts to understand the GDPR, the way it is organized and its role as a European regulation.

The annotators can be, for example, law students or linguists with a minimum knowledge of the language of law.

Appendix 2 : Technical details

XML annotation

The XML annotation of a text consists of inserting annotations, that is to say start and end markers surrounding the text that must be annotated. The markers are called *tags* and together with the

enclosed set form an *element*[7]. In addition to its name, the start tag can have attributes that specify the marking. The *content* of an XML element refers to text and/or elements that are surrounded by its start and end tags. An alternate notation allows merging start and end tags for items with empty content[8]. An XML element can be entirely contained within another or be separate from it, but overlapping is prohibited[9].

To standardize annotations and reduce errors, a *schema* (described in an .xsd or .xd file) can specify the available tags and attributes, as well as legal inclusions between elements, along with their order of inclusion. 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: the names of the same vocabulary receive the same prefix (ending with ":") which abbreviates a standard URI.

Structural annotation

The text of the GDPR is published in an XML format whose schema[10] 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.

The root element is `ACT`. It comprises, after the publication references and the title, a preamble (`PREAMBLE`) including visas and recitals, then the operative part of the regulation (`ENACTING_TERMS`) on which this guide focuses.

The basic overall XML structure of GDPR is as follows in its minimal form:

```
<ACT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:nonamespaceschemaLocation="http://formex.publications.europa.eu/schema/formex/formex-05.55-20141201.xd"
     xmlns:fmx="http://opoce"
     <BIB.INSTANCE> ... </BIB.INSTANCE>
     <TITLE> ... </TITLE>
     <PREAMBLE> ... </PREAMBLE>
     <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

---

[7] `<XXX>xxx xxx</XXX>` where `XXX` is the name of the element, and "`xxx xxx`" is the annotated text.
[8] `<XXX/>` is equivalent to `<XXX></XXX>`.
[9] One can have `<A>... <B>...</B>... </A>` or `<A>... </A>... <B>...</B>` but not `<A>... <B>...</A>...</B>`.
[10] http://formex.publications.europa.eu/schema/formex-05.55-20141201.xd

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 sub-paragraphs, or directly a sequence of sub-paragraphs[11]. A paragraph (PARAG) is numbered while the sub-paragraphs (ALINEA) of the GDPR directly contain text. In the sub-paragraphs, the text is possibly formatted by elements (P and LIST), the schema allowing fairly complex inclusions.

ARTICLE and PARAG have a unique IDENTIFIER attribute formed from the number of the article and the paragraph and which allows to refer to it in the annotations. On the other hand, their superstructures (that is to say the divisions corresponding to the chapters and sections) and the sub-structures of the paragraphs (ALINEA, P and LIST) do not have an identifier and referring to them precisely is therefore not possible.

Examples

```
<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>
        <ALINEA>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.</ALINEA>
  </PARAG>
  <PARAG IDENTIFIER="001.002">
    <NO.PARAG>2.</NO.PARAG>
     <ALINEA>This Regulation protects fundamental rights and freedoms
of natural persons and in particular their right to the protection of
personal data.</ALINEA>
  </PARAG>
```

This example shows the beginning of article 1 in the original xml version of the text with its structure annotations. The article has a title (“Article 1”), a sub-title (“Subject-matter and objectives”) and 3 numbered paragraphs (the first two are reproduced here).

Semantic annotation

Semantic annotations are inserted into a text which is already structured by XML markers. They are inserted at the level of the sub-paragraphs, of their relevant sub-structures (P or TXT), in titles or visas.

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

All XML elements forming the annotation semantic layer include an identifier or refer to an identifier with the exception of sub-fragments.

The **fragments** are annotated as elements of different types (`OBLIGATION`, `PERMISSION`, etc.). Each fragment has a unique identifier[12]. A fragment cannot be inserted into another fragment. A fragment can have relational attributes.

Some **sub-fragments** can be annotated as particular elements without an identifier or an attribute. They are interpreted with regard to the fragment which encompasses them[13].

Entities are elements that play a role in the provisions of the regulation. The Entity Dictionary has a list of entities that each have a unique identifier[14]. In the text of the regulation, the mentions of these different entities are framed by start and end tags which refer to the entity of which they are the mention.

Relationships are marked by attributes associated with fragments that refer either to other fragments or textual elements (relations between fragments) or to entities (roles). The relationships between fragments and sub-fragments are simply marked by the inclusion of the sub-fragment in the fragment.

Examples

The examples below illustrate the use of identifiers for textual elements and entities.

```
<leg:EXCEPTION IDENTIFIER="011.002.002" except="015 016 017 018 019
020">In such cases, Articles 15 to 20 shall not apply
<leg:EXCEPT>except where the <leg:PERSON ref="p_PC">data
subject</leg:PERSON>, for the purpose of exercising his or her rights
under those articles, provides additional information enabling his or
her identification</leg:EXCEPT>.</leg:EXCEPTION>
```

The element `011.002.002` is introduced as an exception to a series of articles, which are numbered `015` to `020` (`except = "015 016 017 018 019 020"`). It also includes an entity mention of the `PERSON` type which refers to (`ref = "p_PC"`) the entity identified in the dictionary by the identifier `id = "p_PC"`.

```
<leg:EXCEPTION IDENTIFIER="006.001.002" except="006.001.001">Point
(f) of the first subparagraph shall not apply to processing carried
```

---

[12] Named `IDENTIFIER` as for other textual elements with an identifier (paragraphs, articles, etc.)

[13] It is impossible to annotate in detail and in a homogeneous way the internal structure of the fragments, but the introduction of sub-fragments nevertheless allows the annotator to underline the particular role played by certain pieces of sentences. This guide recommends annotating sub-fragments that introduce exceptions (see Sub-fragments) but only those

[14] Named `id`.

```
out by public authorities in the performance of their
tasks.</leg:EXCEPTION>
```

Element `006.001.002` is also annotated as an exception to fragment `006.001.001` (the fragment designated as “first sub-paragraph” of the first paragraph of Article 6, in which the text is) taken as a whole (`except = "006.001.001"`) and not an exception to “point f)” of this sub-paragraph as specified in the text, because the points are not textual elements associated with identifiers.

Organization of XML and XSD files

Provided files

Semantic annotations are described in the `GDPR_semantics.xsd` schema, which is automatically included in the `GDPR_S&S_integration.xsd` schema. This last also includes the description of structural annotations.

To integrate the semantic vocabulary and the vocabulary of structure, we use namespaces: the structure vocabulary is not prefixed, while the semantic annotations are prefixed by `leg:`[15], which makes it possible to distinguish the two vocabulates without further inspection.

When you want to read, validate, complete or modify the annotations of the RGPD, a set of files must be installed in the working directory:

- The document which is both structurally and semantically annotated is `GDPR_EN_S&S_annotation.xml`. This is the main file. The original text is enriched with semantic annotations without the structure annotations being modified. It is also associated with dictionaries that are included automatically. The `actorsDictionary.xml` file contains the identifiers of the actors (`PERSON` and `LEGAL_ENTITY`), and `conceptsdictionary.xml` contains those of the concepts. It can be convenient in some annotation steps to keep them under the eyes.

- Only one schema is referenced in the main XML file: `GDPR_S&S_integration.xsd`, which incorporates the description of the structural and semantic annotations and which is dynamically constructed by directly or indirectly using three other files: `GDPR_original_structure.xsd` (the original schema, which describes the vocabulary and grammar of structure annotations), `GDPR_structure_modified.xsd` (an intermediate transformation of the previous one) and `GDPR_semantics.xsd`, which describes the vocabulary and the rules of the semantic annotation. The four files must be in the working directory.

[15] 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.

The dictionary

The dictionary is an XML element without a textual content that is added to the GDPR to list all the entities introduced in the annotation of the regulation and associate them with unique identifiers. It contains entities actually mentioned in the text of the regulation and also a small set of abstract entities used for reference only. To each entity corresponds an entry into the dictionary which specifies its type and possibly one or more text forms (or "labels") under which it can be mentioned in texts.

In practice and for reasons of convenience, the dictionary may be divided into several parts, which amounts to associating several dictionaries with the same annotated document; but the constraint on the uniqueness of the introduced identifiers remains. The repartition into two files provided initially is relatively contingent, and you can modify it as needed.

Syntax

Syntax of a dictionary obeys the following pattern:

```
<leg:DICTIONARY>
(Entity_entry )+
</leg:DICTIONARY>
```

The content of the `DICTIONARY` element is a list of entries that are typed and that introduce unique identifiers of entities.

The `DICTIONARY` element takes no attribute.

Implementation

For reasons of readability, dictionaries, regardless of their number, are all declared in the same place, at the top of the ACT element, root of the document, but they can be implemented in two different ways: either the implementation takes the place of the declaration - In other words, the `DICTIONARY` element is directly included in the XML file at the head of the `ACT` element - or the declaration is an inclusion instruction (`<xi: include>`) of an external dictionary.

The pattern below contains these two modes of association of dictionaries to the XML document that is annotated:

```
<ACT  … >
  (<leg:ABSTRACT_PIECE_OF_TEXT IDENTIFIER="identifier"/>)*
  (<xi:include href="dictionary_file_location"
      xpointer="element(/1/1)"
      xmlns:xi="http://www.w3.org/2001/XInclude"/>|
  <leg:DICTIONARY> (Entity_entry )+ </leg:DICTIONARY>)*
  <BIB.INSTANCE> … <BIB.INSTANCE
  <TITLE> … <TITLE>
```

```
  <PREAMBLE> … <PREAMBLE>
  <ENACTING.TERMS> … <ENACTING.TERMS>
  <FINAL> …… </FINAL>
</ACT>
```

More specifically, the statement of an external dictionary is done by adding a `<xi:include>` sub-element that specifies the external XML file containing the dictionary to include (`href="`*`dictionary_file_location`*`"`) and the dictionary position in the XML tree of this file (`xpointer="element(/1/1)"`)[16]. In the external file, the external dictionary is contained in a `VOCAB` element which is the root of the file and must itself define the namespace of the semantic vocabulary (as the `ACT` element of the RGPD).

```
<VOCAB xmlns:leg="http://www.lipn.univ-paris13.fr/rcln/legal">
      <leg:DICTIONARY>
      (Entity_entry )+
      </leg:DICTIONARY>
</VOCAB>
```

Structure of the provided GDPR file

Once annotated, the GDPR has the following structure:

```
<ACT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://formex.publications.europa.eu/schema
/formex GDPR_S&amp;S_integration.xsd"
      xmlns:fmx="http://opoce"
      xmlns="http://formex.publications.europa.eu/schema/formex"
      xmlns:orig="http://formex.publications.europa.eu/schema/formex"
      xmlns:leg="http://www.lipn.univ-paris13.fr/rcln/legal">
      <leg:ABSTRACT_PIECE_OF_TEXT IDENTIFIER="UNDEFINED"/>
      <xi:include href="actorsDictionary.xml" xpointer="element(/1/1)"
xmlns:xi="http://www.w3.org/2001/XInclude"/>
      <xi:include href="conceptsDictionary.xml"
xpointer="element(/1/1)" xmlns:xi="http://www.w3.org/2001/XInclude"/>
      <BIB.INSTANCE> ... </BIB.INSTANCE>
      <TITLE> ... </TITLE>
      <PREAMBLE> ... </PREAMBLE>
      <ENACTING.TERMS> ... </ENACTING.TERMS>
      <FINAL> ... </FINAL>
</ACT>
```

Additions (in bold in the structure above) concern:

[16] The `xpointer="element(/1/1)` indication specifies that the element to include is the first son of the 1st element (the root) of the external file.

- The reference to the schema integrating structural and semantic annotations (`GDPR_S&amp;S_integration.xsd`),
- The specification of a default name space and the association of the prefix `orig` with the same namespace (which is also the URI of the source document),
- The association of the `leg` prefix with the namespace of semantic annotations (`xmlns:leg="http://www.lipn.univ-paris13.fr/rcln/legal"`),
- The creation of the `UNDEFINED` Identifier by an occurrence of the abstract textual element (`<leg:ABSTRACT_PIECE_OF_TEXT IDENTIFIER="UNDEFINED"/>`),
- The declaration of two dictionaries that are implemented in autonomous files.

Patterns describing the schema of semantic annotations

The semantic annotation of the RGPD must comply with the syntactic rules in the XML schema (`GDPR_Semantics.xsd` file) to which the annotator can refer if he is familiar with the XML schema syntax. To facilitate reading, this document presents, for each XML element, the annotation rules in form of a pattern showing how the element is used (in the .xml file), pattern which is more intuitive to carry out than the rules listed in the schema (.xsd file).

For any XML element of the semantic layer, the XML transposition of the schema is described by the following generic pattern:

<leg:*ELEMENT_NAME* (*attribut_name*="(*value*|*list_of_values*)")*
(*attribut_name*="(*value*|*list_of_values*)")*>*Content*</leg:ELEMENT_NAME>**

which indicates that an element of this semantic layer must have a name prefixed by `leg`:, that it has a possibly empty list of mandatory attributes and optional attributes as well as content. Any attribute is associated with a value that can be a single value or a list of values.

Each of the XML elements of the semantic annotation layer is described by a specific pattern that respects the same annotation principles:

- The text in bold corresponds to compulsory parties;
- The text in lean font corresponds to optional parts;
- The black text must be identically copied;
- Italic and red text should be replaced (in the example above, by an element name, an attribute name, a value or a list of values, or a content);
- The underlined and blue characters are not part of the annotation but of the metalanguage used to describe the annotation rules in XML:

- Parentheses ("(" and ")") mark the boundaries of groups;
- the vertical bar ("|") marks an alternative between the different values of a group[17];
- The asterisk ("*") marks the fact that the preceding group can be repeated a number of times (including 0 times);
- The sign of the addition ("+") marks the fact that the preceding group can be repeated any number of times but at least once.

Let's take an abstract example with the `XXX` element. The pattern below indicates that

- The `XXX` element must necessarily have an `IDENTIFIER` attribute, a `type` attribute and a `bearer` attribute (text in fat) but that it can also have `rel` and `except` attributes (text in lean font);
- The `IDENTIFIER` attribute takes as a value an identifier (*identifier*) to be specified;
- `bearer`, `rel` and `except` attributes take as value an identifier (*identifier*) to be specified or a list of identifiers to be specified and separated by spaces (*identifier*(*identifier*)*);
- The `type` attribute takes a value to be chosen from the group of 3 values associated with the attribute ((**v1**|**v2**|**v3**));
- The content of the element is formed of a mixture of text and XML elements.

<leg:XXX IDENTIFIER="*identifier*" type="(v1|v2|v3)" bearer="*identifier*(*identifier*)*" rel="*identifier*(*identifier*)*" except="(*identifier*(*identifier*)*"**>*Mixed content*</leg:XXX>**

The XML excerpt below complies with this annotation rule: the `XXX` element has four attributes, the three mandatory attributes (`IDENTIFIER`, `type` and `bearer`) and a `rel` attribute but no `except` attribute. All attribute values are identifiers except for the `type` attribute that takes the `v2` value and the `bearer` attribute that is associated with a list of 2 identifiers (`i2 j2`). Finally, the content of the element is a simple sequence of characters (`xxx xxxx xxxx`).

```
<leg:XXX IDENTIFIER="i1" type="v2" bearer="i2 j2" rel="i3">xxx xxxx xxxx</leg:XXX>
```

[17] The notation (a|b) means either a or b.