DALF guidelines for the description and encoding of modern correspondence material

Version 1.0

Edward Vanhoutte & Ron Van den Branden

Centrum voor Teksteditie en Bronnenstudie.

Gent

2003

TABLE OF CONTENTS

APPENDICES

1. Introduction

The availability of correspondences of authors and composers with their family, friends and colleagues, musicians, critics, illustrators, publishers, etc. is of great importance to literary scholars as well as to (art) historians, linguists, cultural sociologists etc. Letters do give an insight in and provide valuable information about, for instance the writing or composing process of an author or a composer, the dating, meaning and reception of a work, and provide further biographical details.

In view of its assignment to study and valorize the Flemish intellectual heritage, the [Centre for Scholarly Editing and Document Studies] (Centrum voor Teksteditie en Bronnenstudie - CTB) has launched the DALF project. DALF is an acronym for "Digital Archive of Letters in Flanders". It is envisioned as a growing textbase of correspondence material which can generate different products for both academia and a wider audience, and thus provide a tool for diverse research disciplines ranging from literary criticism to historical, diachronic, synchronic, and sociolinguistic research. The input of this textbase will consist of the materials produced in separate electronic edition projects. The DALF project can be expected to stimulate new electronic edition projects, as well as the international debate on electronic editions of manuscripts.

In order to ensure maximum flexibility and (re)usability of each of the electronic DALF editions, a formal framework is required that can guarantee uniform integration of new projects in the DALF project. Therefore, the project is from the start aimed at adherence to international standards for electronic text encoding. An important formal standard used in the DALF project is [XML] , that enables the definition of structural text-grammars as Document Type Definitions (DTD). Also in the construction of such a DTD that is suitable for scientific markup of correspondence material, we tried to align with international efforts to define markup schemes. Without going into detail here, the insights and practices presented in international projects like [TEI] (Text Encoding Initiative), [Master] (Manuscript Access through Standards for Electronic Records), and [MEP] (Model Editions Partnership) were taken into consideration for the implementation of following requirements in a DTD for correspondence material:

1. First, the DTD should be designed for the transcription of primary source material, from which letter editions can be generated. In view of this requirement, the work by the Model Editions Partnership proved of little use, since it is aimed at the digitisation of existing letter editions in print.
2. Second, the DTD should allow to store detailed metadata about the transcribed document. This is the point where Master proved useful.
3. Third, the DTD should be able to cater for letter-specific features, such as the information about or written on the envelope, the postscript etc.
4. Fourth, the DTD should allow for a general application to letter

transcriptions and editions and should not restrict itself to the specific corpus of letters we are currently working on.

The TEI encoding scheme provides an excellent starting point for many of the features one would like to encode in letters. Yet, letters have a number of specific elements that require additional encoding means. Since the TEI scheme can be extended when necessary, this is the approach taken for the development of the DALF DTD.

This document explains in detail the letter-specific elements of the DALF DTD, and illustrates their use with a discussion and examples. It will not provide an introduction to the syntax and use of the XML markup language. The interested reader can find a list of introductory materials on XML on [the Cover pages] , and an excellent introduction on the use of XML for humanities computing in chapter [2. A Gentle Introduction to XML] of the TEI P4 Guidelines. Neither does it provide in-depth coverage of all standard TEI elements in the DALF DTD; it assumes instead that the reader is familiar with those, and with the organisation of the TEI tagset. For full and excellent documentation of the TEI tagset the reader is referred to [the TEI P4 Guidelines] .

In section 2 is explained how the DALF DTD extends the TEI scheme, and how both relate to each other. The 3rd section gives a brief overview of the major structural elements of DALF encoded letters. Of these, the specific DALF elements for the header are explained first in section 4. Later, specific DALF text elements are discussed in section 5. Section 6 analyses the problematic area of encoding physical and logical structures with an TEI-based encoding scheme in XML, and the provisions made in the DALF DTD to minimise those problems. Section 7 concludes with a discussion of modifications made to specific TEI element classes, for cases where the distribution of regular TEI elements was considered too narrow for the transcription of primary manuscript materials.

2. The DALF DTD

Many of the features encountered in letters can be encoded with the TEI scheme. However, the TEI tagset has some generally acknowledged lacunae regarding the encoding of primary source material. Chapter [18. Transcription of Primary Sources] of TEI P4 concludes with a hint at future expansion of the TEI tagset in this respect. Apart from items that are applicable to primary manuscripts in general, there are some aspects of a collection of digitally encoded letters that are not covered by the TEI encoding scheme. Fortunately, the option is provided to extend the TEI scheme.

In doing so, we opted for a close structural and terminological resemblance with the TEI tagset when constructing the DALF DTD. There are a number of reasons for this approach:

- To guarantee compatibility with TEI documents, and thus provide users that are familiar with the TEI scheme with an intuitive instrument to mark up correspondence material.
- To present DALF as a true TEI extension, that may enable a substantial evaluation in the light of the TEI tagset, and maybe one day be integrated as an auxiliary tagset for letters in the TEI tagset.
- To guarantee that DALF encoded letters are TEI-conformant documents, as specified in chapter [28. Conformance] of the TEI P4 Guidelines).

Therefore, the DALF DTD has been specified as a customisation of the TEI DTD in the way specified in chapter [29. Modifying and Customizing the TEI DTD] of the TEI P4 Guidelines. The extensions and modifications are encoded in two separate files, [DALFExtns.ent] (for selection and modification of existing TEI tags and element classes) and [DALFExtns.dtd] (for declaration of new elements and attributes), which were input to the [Pizza Chef] program on the interactive TEI website, producing the [DALF.dtd] file. Besides the declaration of new elements and attributes in those modification files, a selection was made of some subsets of standard TEI elements. Elements from the TEI tagset are included in the DALF tagset along the following parameters:

- a ‘mixed base’ tagset was selected, consisting of the **prose** and **drama** bases
- elements from the additional tagsets **Linking**, **figures**, **Analysis**, **transcr**, **textcrit**, and **names.dates** were selected
- the entity sets **ISOlat1**, **ISOlat2**, **ISOnum** and **ISOpub** were selected

[1]

The practical application of the DALF tagset will require choices of the encoders, regarding the level of detail in the meta-descriptions, and between alternatives to

mark up certain phenomena in the source material. These guidelines are in the first place conceived as a descriptive presentation of the provisions made in the DALF tagset to mark up correspondence material. Although alternative encoding approaches are signalled in the guidelines, it is left up to the encoders to develop a preferred practice. To reduce contingency in the encoding of different letters and/or editions, the encoder should

- keep a consistent approach to encode the same phenomena. Minimally within each letter separately, but preferably within all letters of an edition.
- document the approach taken within the **<encodingDesc>** element in the header of each letter.

.....

[1] Concerning the elements included from the additional TEI tagsets: all elements from the **Linking**, **figures**, **transcr**, and **textcrit** additional tagsets were included. Of the **Analysis** additional tagset, only <**interp**> and <**interpGrp**> were included. Of the **names.dates** additional tagset, only <**placeName**>, <**settlement**>, <**region**>, <**country**>, <**bloc**>, and <**distance**> were included.

3. Overview of DALF document structure

To start the presentation of the DALF DTD, this section gives a brief overview of the structure of DALF letters. Like any other TEI document, a DALF letter consists of 2 major constituents: a header part containing meta-data about the letter, and a text part, containing the encoded text. Furthermore, an overview is given of all global attributes that can occur on all elements of the DALF DTD.

3.1 Minimal DALF header

The distinguishing feature of a header for DALF letters is the mandatory presence of a **<letDesc>** element in **<sourceDesc>**. A detailed account of the DALF header is given in section 4. The DALF header (p. 11) . The following example incorporates the further requirements for the header of the TEI scheme (see [the TEI P4 Guidelines]), to give an impression of the elements that must minimally be present in the DALF header:

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>61/ 03.02.1945  Stijn Streuvels to Maurits De Meyer. Ingooigem</title>
    </titleStmt>
    <publicationStmt>
      <publisher>Centrum voor Teksteditie en Bronnenstudie / Centre for Scholarly Editing and
Document Studies</publisher>
      <pubPlace>Gent</pubPlace>
      <date value="2002">2002</date>
      <availability status="restricted">
        <p>&copy; Copyright 2002, CTB</p>
      </availability>
    </publicationStmt>
    <sourceDesc>
      <letDesc>
        <letIdentifier>
          <country>Belgium</country>
          <settlement>Antwerp</settlement>
          <repository>AMVC</repository>
          <collection>S 935/B2</collection>
          <idno>171373/2882</idno>
        </letIdentifier>
        <letHeading>
          <author attested="yes">Stijn Streuvels</author>
          <addressee attested="yes">Maurice De Meyer</addressee>
          <placeLet attested="yes">Ingooigem</placeLet>
          <dateLet attested="yes">1945-02-03</dateLet>
        </letHeading>
        <physDesc>
```

```
            <type>postcard</type>
            <support>
              <p>cardboard postcard, writing in ink on one side only</p>
            </support>
            <extent>
              <dimensions>
                <height units="mm">140</height>
                <width units="mm">90</width>
              </dimensions>
            </extent>
          </physDesc>
          <envOcc occ="yes" />
        </letDesc>
      </sourceDesc>
    </fileDesc>
</teiHeader>
```

3.2 DALF document structure

The simplest type of letter is written by only one author, by and large during a singular time span. This should be encoded with a **<text>** element, in which present envelopes can be encoded with the **<envelope>** element at the same structural level as **<body>**. The letter itself is embedded in **<body>**, with possibilities to encode formal opening and closing constructions in separate **<opener>** and **<closer>** elements (the latter possibly followed by **<ps>** elements for postscripts). The body text is preferably enclosed in **<p>** tags. Editorial notes and lists of links can be listed in a **<back>** part.

```
<TEI.2>
  <teiHeader>...</teiHeader>
  <text>
    <envelope>...</envelope>
    <body>
      <opener>
        <address>...</address>
        <dateline>...</dateline>
        <salute>...</salute>
        ...
      </opener>
      <p>...</p>
      <closer>
        <salute>...</salute>
        <signed>...</signed>
        ...
      </closer>
      <ps>...</ps>
    </body>
    <back>
      <note>...</note>
      <join>...</join>
```

```
      ...
    </back>
  </text>
</TEI.2>
```

For letters in which substantial parts can be distinguished, those parts can be encoded in separate ** ** elements, with a **type** value of 'letPart'. Specific meta-information concerning those different parts constituting the letter can be provided in the header under different **<letPart>** elements. Groups of physically distinct letters can be encoded using the **<group>** tag, with a distinct **<text>** element for each letter. The information for those letters can be given in the header by using different **<sourceDesc>** elements for the letters concerned. As the additional TEI tagset for linking was included in the DALF DTD, it is possible to link different text parts with a **corresp** attribute to the id-values of their representations in the header. The following table summarises the correspondence between header descriptions and text elements for different kinds of letter structure:

text structure	header element	text element
simple letter, no subdivisions	nothing special inside **<letDesc>** (see 4. The DALF header (p. 11))	nothing special; basic structure
composite letter with subdivisions	specific characteristics inside distinct **<letPart>** elements (see 4.8 Distinct letter parts: **<letPart>** (p. 43))	separate ** ** elements (see [the TEI P4 Guidelines])
group of letters	specific characteristics inside distinct **<sourceDesc>** elements (see [the TEI P4 Guidelines])	**<group>** element (see [the TEI P4 Guidelines]) containing separate **<text>** elements (see [the TEI P4 Guidelines])

Note that just like all other forms of encoding, the structural encoding one applies to a letter is a direct consequence of his/her theoretical conception of what 'is' a letter. For example, when a letter is written by one author on the recto side, and continued by another on the verso side, should those be encoded as separate parts inside one letter, or as two distinct letters? Although it is difficult to draw hard and fast rules regarding this question, encoders using the DALF tagset are encouraged to hold a logical-functional view in this matter. This means that the functional autonomy of these parts should be used as a criterium. When there is evidence that both sides clearly have a common communicative purpose, they can be considered as one composite letter. Otherwise they can be considered as a group of two separate

letters. It must be clear that choices like these always are interpretative. In order to maintain consistent encoding, encoders should explicate their presuppositions about the letter governing their encoding practice, in the **<encodingDesc>** element inside the **<teiHeader>** element (see [the TEI P4 Guidelines]).

3.3 Global attributes

All elements in the DALF DTD can have the same basic set of attributes, possibly expanded with specific ones (explained in these guidelines in the discussion of the elements). Since a number of additional TEI tagsets were selected in the composition of the DALF tagset, the set of global attributes for each DALF element consists of the following:

from the core tag set (see [the TEI P4 Guidelines]):

id
Provides a unique identifier for the element bearing the ID value. This is an optional attribute of value type CDATA. Values must be unique in the document and start with a letter or the underscore character ('_'), and contain no characters other than letters, digits, hyphens, underscores, full stops, and certain combining and extension characters.

n
Gives a number (or other label) for an element, which is not necessarily unique within the document. This is an optional attribute of value type CDATA.

lang
Indicates the language of the element content, usually using a two- or three-letter code from ISO 639. This is an optional attribute with as value type IDREF. Its value must thus be an identifier of a **<language>** element supplied in the header (see [the TEI P4 Guidelines]).

rend
Indicates how the element in question was rendered or presented in the source text. This is an optional attribute with as value type CDATA. The encoder can develop any convenient typology for the indication of presentational aspects.

from the additional tag set for linking and segmentation (see [the TEI P4 Guidelines]):

corresp
Points to elements that correspond to the current element in some way. It is an optional attribute with as value type IDREFS. This means that its value must be a space-separated list of one or more valid identifiers of other elements in the document.

synch
Points to elements that are synchronous with the current element. It is an optional attribute with as value type IDREFS. This means that its value must be a space-separated list of one or more valid identifiers of other elements in the document.

sameAs
Points to an element that is the same as the current element. It is an optional attribute with as value type IDREF. Its value must thus be a valid identifier of another element in the document.

copyOf
Points to an element of which the current element is a copy. It is an optional attribute with as value type IDREF. Its value must thus be a valid identifier of another element in the document.

next
Points to the next element of a virtual aggregate of which the current element is part. It is an optional attribute with as value type IDREF. Its value must thus be a valid identifier of another element in the document.

prev
Points to the previous element of a virtual aggregate of which the current element is part. It is an optional attribute with as value type IDREF. Its value must thus be a valid identifier of another element in the document.

exclude
Points to elements that are in exclusive alternation with the current element. It is an optional attribute with as value type IDREFS. This means that its value must be a space-separated list of one or more valid identifiers of other elements in the document.

select
Selects one or more alternants; if one alternant is selected, the ambiguity or uncertainty is marked as resolved. If more than one alternant is selected, the degree of ambiguity or uncertainty is marked as reduced by the number of alternants not selected. It is an optional attribute with as value type IDREFS. This means that its value must be a space-separated list of one or more valid identifiers of other elements in the document.

from the additional tag set for simple analysis (see [the TEI P4 Guidelines]):

ana
Indicates one or more elements containing interpretations of the element on which the ana attribute appears. It is an optional attribute with as value type IDREFS. This means that its value must be a space-separated list of one or more valid identifiers of other elements in the document.

4. The DALF header

The DALF textbase may develop to a collection of thousands of electronic texts. Although these textual materials will all be letters and thus share a lot of characteristics, the nature of the letter genre (letters are highly authorial ego-documents) implies they may differ substantially. The best way to guarantee consistency and thus usability of the texts within the whole, is the encoding of a rich amount of abstracted meta-information for each letter. The TEI scheme provides useful elements for such abstractions in the header part of the document. However, a textbase of correspondence material calls for more specific provisions to capture the indication of letter-specific and archival metadata, than those provided in the TEI scheme. The DALF DTD therefore specifies a number of extensions to the TEI header. Many of those extensions are inspired by the header elements of the DTD developed for the [Master] project (Manuscript Access through Standards for Electronic Records), which was "intended primarily for the detailed cataloguing of medieval and early modern manuscript materials in the Western European tradition". The envisioned outline of DALF differs, however, in two major aspects from the goals of the Master project:

- DALF will be a textbase of *electronic editions* of primary sources (whereas the Master project aims at a database of manuscript description records)
- DALF will focus on an archive of digital editions of *modern correspondence* (whereas the Master project is aimed at documenting medieval manuscripts)

In view of these differences, the DALF header is conceived as a standard TEI header with extension elements that are organised similarly to the Master header, but have been reformulated to express the particular nature of an archive of letters. The following principles have been adhered to in the design of the DALF header:

- In order to keep the semantics transparent, all letter-specific elements in the header have names starting with "let-".
- In order to ensure consistent encoding of DALF documents and to facilitate their integration into a searchable electronic textbase, we opted for a fairly strict design of the header. This resulted in several mandatory elements or mandatory choices between alternatives.
- To ensure flexibility, optional **<note>** elements are allowed after mandatory contents.

The main feature of the DALF header[2] is the **<letDesc>** element that is added to the standard TEI **<sourceDesc>** element. It is a mandatory element that groups together all letter-specific metadata for a DALF document.

<letDesc>
Groups together all letter-specific metadata for a DALF document.

Apart from the global TEI attributes (described in 3.3 Global attributes (p. 8)), it can carry one special attribute:

status
This optional attribute indicates the compositional status of a letter. Possible values are: ‘uni’ for unitary texts; ‘compo’ for composite texts; ‘frag’ for texts that consist only of (a) fragment(s); ‘def’ for defective texts of which only a small piece is missing. The value ‘unk’ is provided for letters where none of the above is known.

The letter description consists of the following elements:

<letIdentifier>
Contains information concerning the identification of the letter within its holding institution.

<letHeading>
Contains a structured description of bibliographical information of a letter.

<physDesc>
Contains a description of the physical appearance of the letter.

<envOcc />
Contains an indication of the presence or absence of an envelope.

<letContents>
Contains a description of the intellectual contents of the letter.

<history>
Contains a description of the history of the letter.

<additional>
Groups additional information about the letter.

<letPart>
Contains metadata about distinct parts of a letter.

<note>
Contains additional information about the letter that is not covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

.....

[2] This section will discuss the specific extensions of the TEI header that are incorporated in the DALF header. This means that some familiarity with the TEI header is assumed from the reader. We considered it no use to repeat the excellent and comprehensive documentation for the TEI header, the interested (or desperate) reader can find at [the TEI P4 Guidelines] . As for the standard TEI elements occurring within specific DALF header elements, a concise explanation is provided, as well as a reference to the full TEI documentation of the elements concerned.

Of these, **<letIdentifier>**, **<letHeading>**, **<physDesc>** and **<envOcc />** are mandatory elements that must occur exactly once. The elements **<letContents>**, **<history>** and **<additional>** are optional and may occur only once. **<letPart>** is an optional element that may occur more than once. The **<note>** element is optional and may only occur after all other header elements. Another restriction is imposed on the order of the elements: they *must* appear in the order specified. This means that a DALF header always consists of a **<letIdentifier>**, followed by a **<letHeading>**, **<physDesc>** and a **<envOcc />**, optionally followed by at most one **<letContents>**, **<history>**, **<additional>**, and zero or more **<letPart>** or **<note>** elements in that order. Here is an illustration of the structure of a minimal DALF header:

```
<teiHeader>
  <fileDesc>
    <titleStmt>...</titleStmt>
    <publicationStmt>...</publicationStmt>
    <sourceDesc>
      <letDesc>
        <letIdentifier>...</letIdentifier>
        <letHeading>...</letHeading>
        <physDesc>...</physDesc>
        <envOcc occ="..." />
        ...
      </letDesc>
    </sourceDesc>
  </fileDesc>
</teiHeader>
```

The following lines demonstrate how the file DALFExtns.dtd redefines the **<sourceDesc>** element to include **<letDesc>**, and how the latter element itself is defined:

```
<!ELEMENT sourceDesc %om.RR;
        (biblStruct?, letDesc, note*)>
<!ATTLIST sourceDesc
        %a.global;
        %a.declarable;>

<!ELEMENT letDesc %om.RR;
        (letIdentifier, letHeading, physDesc, envOcc, letContents?, history?,
        additional?, letPart*, note*)>
<!ATTLIST letDesc
        %a.global;
        status (uni | compo | frag | def | unk) #IMPLIED>
```

In what follows, all of these elements will be explained in full detail. The sections are organised in a uniform way: first, their purpose is explained and a short description of their attributes and contents is given, as well as their formal declaration in the DALF DTD. When applicable, the child elements are explained in full detail in different

subsections, following the same structure.

4.1 The letter identifier: <letIdentifier>

Most of the letters that will constitute the DALF database will be unpublished primary manuscripts that are stored in private or public collections, conserved at particular places. Therefore, there is need for a more detailed level of cataloguing description than for published source materials, providing researchers with a uniform and accurate system of archival reference for each letter.

Therefore, the **<letIdentifier>** element is specified as the first mandatory element that appears in the **<letDesc>** element. Here, the encoder must supply information regarding the identification of the document within its holding institution. This identification starts with a group of elements describing the hierarchical location path, which must minimally contain an indication of the country, settlement of the institution and the name of the repository where the document is located. When appropriate, intervening levels in the location path can be identified for the region where the repository is located, and the institution where the repository resides. A second group of elements of the letter identifier describe the location of the document within its holding institution. The collection to which the document belongs, as well as the identification that is given to the document within its repository are required. When the document is known under an alternative name, this can also be given.

The **<letIdentifier>** has no other attributes than the global ones (see 3.3 Global attributes (p. 8)). The identification for a DALF letter is captured in following elements (in the order specified):

<country>
Describes the country where the document is located. This is a TEI element for the name of a geo-political unit larger than or administratively superior to a region. (see [the TEI P4 Guidelines])

<region>
Describes the region where the document is located. This is a TEI element for the name of geo-political unit which is larger or administratively superior to the settlement and smaller or administratively less important than the country. (see [the TEI P4 Guidelines])

<settlement>
Describes the settlement where the document is located. This is a TEI element for the smallest component of a place name expressed as a hierarchy of geo-political or administrative units. (see [the TEI P4 Guidelines])

<institution>
Contains the name of an organisation within which a manuscript repository is located.

<repository>
Contains the name of a repository (usually a distinct physical building) within which manuscripts are stored, forming part of an institution.

<collection>
Contains an identification of a collection of documents of which the letter forms part, not necessarily located within a single repository.

<idno>
Contains the identification number that is used to identify the document. This is a standard TEI element (see [the TEI P4 Guidelines])

<altName>
Contains any form of alternative identifier used for a document. This may be a nickname or a former identifier.

<note>
Contains additional remarks that are not covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

Of these, **<country>**, **<settlement>**, **<repository>**, **<collection>** and **<idno>** are mandatory and must occur exactly once. Optional elements that may only occur once are **<region>** and **<institution>**. The other elements, **<altName>** and **<note>** are optional and may occur more than once. The following example illustrates the minimal letter identifier for a DALF letter that is located at the AMVC in Antwerp, Belgium:

```
<letIdentifier>
  <country>Belgium</country>
  <settlement>Antwerp</settlement>
  <repository>AMVC</repository>
  <collection>S 935/B2</collection>
  <idno>171373/2882</idno>
</letIdentifier>
```

The **<letIdentifier>** element is declared in the file DALFExtns.dtd as follows:

```
<!ELEMENT letIdentifier %om.RR;
        (country, region?, settlement, institution?, repository, collection, idno, altName*, note*)>
<!ATTLIST letIdentifier
        %a.global;>
```

4.1.1 The macro-location path of a letter: <country>, <region>, <settlement>, <institution> and <repository>

These elements give a top-down description of the location path of a letter, from the country where the holding institution is located to the most specific unit within that institution that can be distinguished for the place where the letter is located.

The elements **<country>**, **<region>** and **<settlement>** are all standard TEI elements carrying the attributes of the **names** and **typed** attribute classes. The elements **<institution>** and **<repository>** are adapted from the Master encoding scheme; since they have similar semantics, they have been uniformed with the TEI elements. This means that they all may contain elements of the **phrase.seq** element class. The elements defined by that class are too numerous to list here; they are listed exhaustively at [the TEI P4 Guidelines] . All elements expressing the macro-location path of a letter may contain the following attributes, apart from the global ones (defined in 3.3 Global attributes (p. 8)):

from the TEI attribute class **names** (defined at [the TEI P4 Guidelines]):

key
Provides an alternative identifier for the object being named, such as a database record key. This is an optional attribute, with CDATA as value type.

reg
Gives a normalised or regularised form of the name used, in the form of CDATA. This is an optional attribute.

from the TEI attribute class **typed** (defined at [the TEI P4 Guidelines]):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

subtype
Provides a sub-categorisation of the element, if needed. It is an optional attribute with as value type CDATA.

The following example illustrates a more extensive location path for the letter of the previous example, that is located in the AMVC in Antwerp, Flanders, Belgium. Here, the region where the settlement is located is given, the institution to which the repository belongs is identified and typed as a municipal department, and the repository is provided with a database record key and a regularised form of its name:

```
<country>Belgium</country>
<region>Flanders</region>
<settlement>Antwerp</settlement>
<institution type="municip_department">Dept. of Culture, Antwerp</institution>
<repository key="loc01_AMVC" reg="AMVC">AMVC</repository>
```

The unique DALF elements **<institution>** and **<repository>** are declared as follows in the file DALFExtns.dtd:

```
<!ELEMENT institution %om.RR;
      (%phrase.seq;)>
```

```
<!ATTLIST institution
        %a.global;
        %a.names;
        %a.typed;>

<!ELEMENT repository %om.RR;
        (%phrase.seq;)>
<!ATTLIST repository
        %a.global;
        %a.names;
        %a.typed;>
```

4.1.2 The micro-location path of a letter: <collection>, <idno> and <altName>

The identification of a letter continues to the micro-level: the identification of the letter *within* its repository. The collection to which the letter belongs must be identified in the **<collection>** element. The identification code that is used within the repository to indicate the letter must be specified in the **<idno>** element. If a letter may be known under another name or identification code, this can be recorded in **<altName>**.

The **<collection>** and **<altName>** elements are closely related to the TEI name class elements. Therefore, they too can contain the elements of the **phrase.seq** class (see [the TEI P4 Guidelines]). Their attributes, apart from the global ones (see 3.3 Global attributes (p. 8)), are:

from the TEI attribute class **names** (defined at [the TEI P4 Guidelines]):

key
Provides an alternative identifier for the object being named, such as a database record key. This is an optional attribute, with CDATA as value type.

reg
Gives a normalised or regularised form of the name used, in the form of CDATA. This is an optional attribute.

from the TEI attribute class **typed** (defined at [the TEI P4 Guidelines]):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

subtype
Provides a sub-categorisation of the element, if needed. It is an optional attribute with as value type CDATA.

The **<idno>** element, which is a standard TEI element (see [the TEI P4 Guidelines]), can contain an identification code in plain PCDATA format, and has one attribute apart from the global ones (see 3.3 Global attributes (p. 8)):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

The following example shows a full micro-identification in the **<letIdentifier>** part of a DALF letter that is member of the Streuvels-Lannoo collection identified by the code S 935/62295, in which it carries the ID 71373/2882, and has a former identification code number and an alternative nickname:

```
<collection>S 935/62295</collection>
<idno>71373/2882</idno>
<altName type="former_id">SL410811/b</altName>
<altName type="nickname">the birthday letter</altName></repository>
```

The unique DALF elements **<collection>** and **<altName>** are declared as follows in the file DALFExtns.dtd:

```
<!ELEMENT collection %om.RR;
        (%phrase.seq;)>
<!ATTLIST collection
        %a.global;
        %a.names;
        %a.typed;>

<!ELEMENT altName %om.RR;
        (%phrase.seq;)>
<!ATTLIST altName
        %a.global;
        %a.names;
        %a.typed;>
```

4.2 The letter heading: <letHeading>

One of the essential characteristics of letters is their close relationship with the particular communicative context in which they are created. Of course, this also holds for published books, that are written by a certain author and at a certain place and time. Yet, as bibliographic references to books show, those particular communicative circumstances of the writing act are deemed less important than the circumstances of publication. In contrast, when referring to letters as unambiguously as possible, one has to include as much of the communicative particularities as possible. Those are so important that they may be considered an essential part of the bibliographical identification of a letter.

Therefore, the **<letHeading>** element is taken as the second mandatory element in the DALF header. The letter heading starts with an identification of the participants in the communication. Those consist of the author(s) and addressee(s) of a letter, and when applicable other persons with any responsibility for some aspects of the contents. The second part of the letter heading covers the communicative situation, in the form of the place and time when the letter was written.

The letter heading has only the global attributes (see 3.3 Global attributes (p. 8) , and contains the following elements in the order specified:

<author>
Identifies an/the author of the letter.

<addressee>
Identifies a/the person to whom the letter was addressed.

<respStmt>
Identifies a person who is responsible for some aspect of the contents. This is a standard TEI element (see [the TEI P4 Guidelines]).

<placeLet>
Identifies the place where the letter was written.

<dateLet>
Identifies the date when the letter was written.

<note>
Contains additional remarks that are not covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

The elements **<author>** and **<addressee>** are mandatory and must occur at least once. The **<respStmt>** element is optional and may as well occur more than once. Both **<dateLet>** and **<placeLet>** are mandatory, and must occur exactly once.

The following example shows a minimal letter identifier for a letter written by Streuvels to De Meyer in Ingooigem on 13 January 1941:

```
<letHeading>
  <author>Stijn Streuvels</author>
  <addressee>Maurice De Meyer</addressee>
  <placeLet>Ingooigem</placeLet>
  <dateLet>1945-01-13</dateLet>
</letHeading>
```

The **<letHeading>** element is declared in the DALFExtns.dtd file in the following way:

```
<!ELEMENT letHeading %om.RR;
        (author+, addressee+, respStmt*, placeLet, dateLet, note*)>
<!ATTLIST letHeading
        %a.global;>
```

4.2.1 The communicative participants: <author>, <addressee> and <respStmt>

The participants in the communicative act of letter writing are the writer(s) and reader(s) involved. Their communicative roles are reflected in the names of the elements **<author>** and **<addressee>**, that the encoder must use to record the author, respectively addressee of the letter. Most often, they are stated explicitly in the letter; otherwise the encoder can use other sources of evidence that can provide a plausible indication of their identities. When a letter is written by several authors, and their contributions are not encoded as separate **<letPart>**s in the header (see 4.8 Distinct letter parts: **<letPart>** (p. 43)), these can be identified in several **<author>** elements. Also, when a letter has been written to more than one addressee, several **<addressee>** elements can be used to enumerate all of them. Optionally, people other than the author that are responsible for some aspects of the contents can be identified in zero or more **<respStmt>** elements.

Note that the identification of multiple authors for a letter may arouse questions regarding the compositional structure of the letter. The encoder must decide whether the letter should be considered a whole, with distinct authors identified internally in several **<author>** elements within **<letHeading>**; or should be considered to have distinct parts, with their characteristics encoded in separate **<letPart>** elements within **<letDesc>** (see 4.8 Distinct letter parts: **<letPart>** (p. 43)).

The **<author>** and **<addressee>** elements are closely related to the TEI name class elements. Therefore, they can contain the elements of the **phrase.seq** class (see [the TEI P4 Guidelines]), and the attributes from the **names** and **typed** attribute classes. Most often, the author and addressee are explicitly identified in a letter. In other cases, the encoder must try to identify them from other pieces of evidence, like the envelope or for example other letters in the correspondence chain. Therefore, the **<author>** and **<addressee>** elements have two additional attributes that can record the status of the evidence on which the identification was based. Together with the **names** and **typed** class attributes, these elements can have following attributes (besides the global ones defined in 3.3 Global attributes (p. 8)):

attested
Indicates the status of the contents of the element on which it appears. This is an optional attribute with one of four predefined values. The encoder can indicate whether the abstraction is made on the basis of evidence inside the letter, with value ‘yes’; additional material accompanying the letter (e.g. the envelope), with value ‘added’; derived from external evidence (e.g. a reply letter), with value ‘no’; or if the feature is unknown, with value ‘unk’.

accepted
Indicates whether the attribution made in the element is generally

accepted. This is an optional attribute with one of following three values. Generally accepted attributions are encoded with value ‘yes’; attributions that are not generally accepted with ‘no’; when no claim can be made regarding this status, the value ‘unk’ is used.

from the TEI attribute class **names** (defined at [the TEI P4 Guidelines]):

key
Provides an alternative identifier for the object being named, such as a database record key. This is an optional attribute, with CDATA as value type.

reg
Gives a normalised or regularised form of the name used, in the form of CDATA. This is an optional attribute.

from the TEI attribute class **typed** (defined at [the TEI P4 Guidelines]):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

subtype
Provides a sub-categorisation of the element, if needed. It is an optional attribute with as value type CDATA.

The **<respStmt>** element can be used to describe any person other than the author proper, that has contributed something to the contents of the letter, be it conceptually or physically. It must be stressed that the tracing of such contributions (non-material ones in particular) is not the most straightforward task, and should be done with meticulous care (as is the case with all aspects of the electronic edition of DALF letters, of course). It may not always be easy to distinguish between elements that are present on the document but are totally unrelated/unimportant to its contents and thus can hardly be considered as *contributions*, or elements that are so important they can be considered as much more than just a contribution and thus be encoded in a different way. As a rule, the encoder can safely use the **<respStmt>** element within **<letHeading>** to identify a person who typed in the letter, or provided some ideas for the contents (supposed this can be traced at all). Yet, the encoder should keep a consistent practice concerning the use of the **<respStmt>** for the identification of persons who made some additions to the letter. For plain additions, their author can also be identified within the **scribe** attribute of the **<hand>** element inside the **<profileDesc>** element of the heading (see [the TEI P4 Guidelines]). For very lengthy and substantial contributions the encoder should consider whether their authors shouldn't better be encoded as separate authors, or even in separate **<letPart>** elements (see 4.8 Distinct letter parts: **<letPart>** (p. 43)). Perhaps the TEI definition of **<respStmt>** (see [the TEI P4 Guidelines]) should give a good indication

regarding its use: '(statement of responsibility) supplies a statement of responsibility for someone responsible for the intellectual content of a text, edition, recording, or series, *where the specialized elements for authors, editors, etc. do not suffice or do not apply*' [italics added].

The attributes of the **<respStmt>** element are those defined as the global attributes (see 3.3 Global attributes (p. 8)). It can contain a standard TEI **<resp>** element (see [the TEI P4 Guidelines]), naming the nature of the responsibility of the person identified in a standard TEI **<name>** element (see [the TEI P4 Guidelines]), and can furthermore contain the elements of the **Incl** element class (see A.2.b.6 Incl (p. 144)).

The following example illustrates the extensive identification of the communicative participants within the **<letHeading>** part of a DALF letter written by Streuvels to De Meyer. Here, it could be stated that Streuvels dictated the letter to his wife, who typed it in. Thus, Streuvels can be considered the intellectual author of the letter, while his wife can be attributed responsibility as scribe. Also, an idea in the letter could be attributed by the encoder to Joris Lannoo. As indicated in the attributes, both sender and addressee are identified from evidence inside the letter:

```
<author attested="yes">Stijn Streuvels</author>
<addressee attested="yes">Maurice De Meyer</addressee>
<respStmt>
  <resp>scribe</resp>
  <name>Alida Staelens</name>
</respStmt>
<respStmt>
  <resp>idea for the suggestion to use an instalment system of publication</resp>
  <name>Joris Lannoo</name>
</respStmt>
```

The unique DALF elements **<author>** and **<addressee>** are declared as follows in the file DALFExtns.dtd:

```
<!ELEMENT author %om.RR;
        (%phrase.seq;)>
<!ATTLIST author
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED
        accepted (yes | no | unk) #IMPLIED>

<!ELEMENT addressee %om.RR;
        (%phrase.seq;)>
<!ATTLIST addressee
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED
        accepted (yes | no | unk) #IMPLIED>
```

4.2.2 The place and time of writing: <placeLet> and <dateLet>

The place and time of writing can be identified respectively with **<placeLet>** and **<dateLet>**. As can be assumed that a letter is written in a short time span and at one place only, these elements can occur only once. Should the encoder wish to make some elaborations regarding this assumption for a certain letter, this can be done in the concluding **<note>** element of the **<letHeading>** element (as is the case for all DALF header elements). Often the place and time of writing are included in the salutation part of the letter; otherwise the encoder can use place and time information from the envelope or from other sources of evidence.

The **<placeLet>** element is closely related to the TEI name class elements. Therefore, it can contain the elements of the **phrase.seq** class (see [the TEI P4 Guidelines]), and the attributes from the **names** and **typed** attribute classes. The source of evidence for the identification of the place of writing may be given in the special **attested** attribute. Apart from the global attributes listed in 3.3 Global attributes (p. 8) , **<placeLet>** can have following attributes:

attested
Indicates the status of the contents of the element on which it appears. This is an optional attribute with one of four predefined values. The encoder can indicate whether the abstraction is made on the basis of evidence inside the letter, with value ‘yes’; additional material accompanying the letter (e.g. the envelope), with value ‘added’; derived from external evidence (e.g. a reply letter), with value ‘no’; or if the feature is unknown, with value ‘unk’.

from the TEI attribute class **names** (defined at [the TEI P4 Guidelines]):

key
Provides an alternative identifier for the object being named, such as a database record key. This is an optional attribute, with CDATA as value type.

reg
Gives a normalised or regularised form of the name used, in the form of CDATA. This is an optional attribute.

from the TEI attribute class **typed** (defined at [the TEI P4 Guidelines]):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

subtype

Provides a sub-categorisation of the element, if needed. It is an optional attribute with as value type CDATA.

The **<dateLet>** element can contain the elements of the **phrase.seq** class that are listed in [the TEI P4 Guidelines] . Apart from the global attributes (see 3.3 Global attributes (p. 8) , the special attribute **attested** is provided to supply the source of the identification of the place of writing:

attested
Indicates the status of the contents of the element on which it appears. This is an optional attribute with one of four predefined values. The encoder can indicate whether the abstraction is made on the basis of evidence inside the letter, with value ‘yes’; additional material accompanying the letter (e.g. the envelope), with value ‘added’; derived from external evidence (e.g. a reply letter), with value ‘no’; or if the feature is unknown, with value ‘unk’.

The following example illustrates the extensive identification of the place and time of writing of a letter written by Stijn Streuvels to Maurice De Meyer in Ingooigem, on 13 January 1945. As indicated in the attributes, the place of writing is derived from evidence on the envelope, and the date of the letter is derived from external evidence.

```
<placeLet attested="added">Ingooigem</placeLet>
<dateLet attested="no">1945-01-13</dateLet>
```

The unique DALF elements **<placeLet>** and **<dateLet>** are declared as follows in the file DALFExtns.dtd:

```
<!ELEMENT placeLet %om.RR;
        (%phrase.seq;)>
<!ATTLIST placeLet
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED>

<!ELEMENT dateLet %om.RR;
        (%phrase.seq;)>
<!ATTLIST dateLet
        %a.global;
        attested (yes | added | no | unk) #IMPLIED>
```

4.3 The physical description: <physDesc>

The description of the physical appearance of the letter forms the third mandatory element of the **<letDesc>** element. As letters can be very different with regard to their physical realisation, the description of physical aspects contains a limited set of

elements for features that are shared by all letters, and additionally provides the possibility to encode a rich array of specific phenomena when they occur. Required elements are a characterisation of the format of the letter, a description of the material on which the letter is written, and an indication of the physical size of the document. Additionally, general aspects of layout can be pointed out, as well as a characterisation of possible fragments of musical notation, a description of possible decorative elements or paraphernalia, and the condition of the letter as a physical object.

The **<physDesc>** element has only global attributes (see 3.3 Global attributes (p. 8)). It captures the physical description of a letter in following elements:

<type>
Gives a formal characterisation of the letter.

<support>
Gives a description of the material on which the letter is written.

<extent>
Describes the approximate size of the electronic text as stored on some carrier medium, specified in any convenient units. This is a standard TEI element (see [the TEI P4 Guidelines]).

<layout>
Contains a description of specific layout aspects of the letter.

<musicNotation>
Describes the type of musical notation that appears in the letter.

<decoration>
Describes decorative items in the letter that have originated within the writing act.

<paraphernalia>
Describes decorative items in the letter that have originated independently of the writing act.

<condition>
Describes the physical condition of the letter.

<note>
Contains additional remarks that are not covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

Of these elements, the required ones (**<type>**, **<support>** and **<extent>**) must occur exactly once. When optional elements are used, they too may occur only once (except for **<note>**). The following example illustrates the minimal encoding of a letter that is written on an single sheet of paper. That paper has an A4 format, and has a letterhead printed on it.

```
<physDesc>
```

```
  <type>letter</type>
  <support>
    <p>single page with pre-printed letterhead, writing on one side only</p>
  </support>
  <extent>
    <dimensions>
      <height units="mm">214</height>
      <width units="mm">276</width>
    </dimensions>
  </extent>
</physDesc>
```

The **<physDesc>** element is declared in the DALFExtns.dtd file in the following way:

```
<!ELEMENT physDesc %om.RR;
        (type, support, extent, layout?, musicNotation?, decoration?, paraphernalia?, condition?,
note*)>
<!ATTLIST physDesc
        %a.global;>
```

4.3.1 Formal characterisation: <type>

As it is imaginable that letters can be written in virtually any format, the formal characterisation of the letter can be given in a prose description. The encoder should develop a consistent typology in order to guarantee retrieval possibilities for this category. Typical values can be ‘letter’, ‘postcard’ etc. When for example for very recent authors and composers it is thought theoretically sound to include saved E-mail messages too in an electronic edition of correspondence, a format like ‘E-mail’ may be specified for **<type>**. The **<type>** element has only the global attributes (3.3 Global attributes (p. 8)).

The following example illustrates the **<type>** encoding for a postcard:

```
<type>
  postcard
</type>
```

The **<type>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT type %om.RR;
        (#PCDATA) >
<!ATTLIST type
        %a.global;>
```

4.3.2 Support material: <support>

The materials that were used to write the letter on must be characterised in one or more **<p>** elements inside **<support>**. This description may cover the material of the support material (e.g. 'paper'), the number of pages and the sides that are used.

The **<support>** element has no other than the global attributes (see 3.3 Global attributes (p. 8)), and can contain following elements:

<p>
Contains the description of the material on which the letter is written. This is a standard TEI element (see [the TEI P4 Guidelines]).

For a letter written on a very thin piece of paper, the following may be a legal encoding:

```
<support>
  <p>very thin, semi-transparent paper</p>
</support>
```

The **<support>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT support %om.RR;
        (p+) >
<!ATTLIST support
        %a.global;>
```

4.3.3 Size: <extent>

The size of a letter must be described in the **<extent>** element, which is a standard TEI element (see [the TEI P4 Guidelines]). This element may hold a plain prose description of the size of the letter, or elements from the element class that is defined in TEI P4 as **phrase.seq** (see [the TEI P4 Guidelines]). A common way to express the size of the document in the header, is to use the **<dimensions>** tag inside **<extent>**. That phrase-level element can provide a structured description of the height and width of the item it represents, as well as a unit of measure.

The following example illustrates how the size of a letter can be described, using the **<dimensions>** element within **<extent>**:

```
<extent>
  <dimensions>
    <height units="mm">214</height>
    <width units="mm">276</width>
  </dimensions>
</extent>
```

4.3.4 Layout aspects: <layout>

Specific layout aspects, like the use of columns and all kinds of outstanding visual features may be expressed in **<p>**s inside **<layout>**.

The **<layout>** element has no other than the global attributes (see 3.3 Global attributes (p. 8)), and can contain following elements:

<p>
Contains the description of the layout of the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

The following example illustrates how the **<layout>** element can be used to record that a letter has 3 columns that read from right to left:

```
<layout>
  <p>written in 3 columns, starting with the rightmost and continuing to the left</p>
</layout>
```

The **<layout>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT layout %om.RR;
        (p+) >
<!ATTLIST layout
        %a.global;>
```

4.3.5 Musical notation: <musicNotation>

Considering the outline of DALF as a Digital Archive of Letters in Flanders, focusing on letters written by authors and composers from the 19th and 20th century, it is very likely that fragments of musical notation can appear in letters. With the **<musicNotation>** element, the encoder can provide a typology of the notation employed. This can be done inside one or more paragraphs.

The **<musicNotation>** element has no other than the global attributes (see 3.3 Global attributes (p. 8)), and can contain following elements:

<p>
Contains the description of the kind of musical notation in the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

For example, a letter containing 2 different instances of musical notation, consisting of 3 and 2 staves respectively, may be encoded as follows:

```
<musicNotation>
  <p>theme of "Struggle for Pleasure" in 3 staves</p>
  <p>theme of "The Scene" in 2 staves</p>
</musicNotation>
```

The **<musicNotation>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT musicNotation %om.RR;
        (p+) >
<!ATTLIST musicNotation
        %a.global;>
```

4.3.6 Decorative elements: <decoration> and <paraphernalia>

Letters can contain all sorts of ‘less-textual’ contents (so called because they have a non-textual nature, but still *are* part of the text). Although it's a theoretical issue whether or not to regard accompanying materials or ‘objets trouvés’ as a part of the letter, the DALF header provides the means to do so. A distinction is made between decorations, i.e. less-textual materials that can be regarded to have originated within the writing act, and paraphernalia, materials that did not originate within the writing act. Decorations can include drawings, schemes etc., whereas paraphernalia can include newspaper articles, hair curls, dried flowers etc. It has to be pointed out that the DALF scheme also provides the means to mark another category of material that has a somewhat closer formal (but not necessarily intellectual) bond with the letter, namely printed text that was already on the paper before the letter was written (e.g. form text), or that has been added afterwards (e.g. stamps). When these elements are not regarded as decorative elements that need meta-description in the header, they can be marked in the document itself with the **<print>** element discussed in 5.4 Pre- and post-printed material: **<print>** (p. 55) .

The inclusion of specific descriptive elements for these less-textual materials in the DALF header is motivated because of the potential importance of decorations and paraphernalia for genetic studies. In view of the size and diversity of the DALF textbase, it could be very interesting to have the means to search the archive for specific instances of less-textual materials.

Both the **<decoration>** and **<paraphernalia>** elements have the same structure. They can either contain a loose prose description in one or more **<p>**s, or a structured description per decorative item concerned. The latter is conceived as a list, consisting of a **<decoList>**, respectively **<paraphList>** element. These must contain one or more **<decoItem>** or **<paraphItem>** elements, one per decorative item in the document. Those list items must contain a description of the decorative item, in **<decoDesc>** or **<paraphDesc>**. Optionally, also text appearing within the decorative item can be recorded in **<decoText>** or **<paraphText>**. Both the description and the recording of the text must go in one or more paragraphs. The following list gives a more formal overview of the contents of the structured descriptions inside **<decoration>** and **<paraphernalia>**. Although both systems are clearly distinct, they do have a similar structure, which is the only reason why they are discussed alongside each other.

<decoList> and <paraphList>
Contains a list of structured descriptions for each of the decorations, respectively paraphernalia in a letter.

<decoItem> and <paraphItem>
Contains a structured description of a decoration, respectively paraphernalia in a letter.

<decoDesc> and <paraphDesc>
Contains a description of a decoration, respectively paraphernalia in a letter, inside one or more **<p>** elements (see [the TEI P4 Guidelines]).

<decoText> and <paraphText>
Contains a rendering of text found as the contents of a decoration, respectively paraphernalia. This has to be done within one or more **<p>** elements (see [the TEI P4 Guidelines]).

<p>
Contains a loose description of the decorations, respectively paraphernalia in the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

All of these elements have only the global attributes (see 3.3 Global attributes (p. 8)). The following example shows how a letter containing two drawings (one of which containing text) and a dried flower attached to it, can be encoded in the DALF header:

```
<decoration>
  <decoList>
    <decoItem id="fig1">
      <decoDesc>
        <p>small black/white drawing of a tree</p>
      </decoDesc>
    </decoItem>
    <decoItem id="fig2">
      <decoDesc>
        <p>small color drawing of suggested page layout, containing some text</p>
      </decoDesc>
      <decoText>
        <p>Eerste paragraaf Hier</p>
        <p>voetnoten!</p>
      </decoText>
    </decoItem>
  </decoList>
</decoration>
<paraphernalia>
  <paraphList>
    <paraphItem id="obj1">
      <paraphDesc>
        <p>dried petal of a poppy</p>
      </paraphDesc>
```

```
    </paraphItem>
  </paraphList>
</paraphernalia>
```

This list construction is modelled after the TEI mechanism of identifying and referring to different document hands (see [the TEI P4 Guidelines]). This means that in the header the definitions discussed above are provided for less-textual elements in the text. When doing so, the encoder must provide each of the **<decoItem>** and **<paraphItem>** elements with a unique value for the **id** attribute. When those instances are encoded in the text, the specific empty DALF elements **<deco />** and **<paraph />** can indicate their occurrence, and refer to the description in the header. These text elements are explained in the next section (5.5 Decorative elements: **<deco />** and **<paraph />** (p. 57)), but the following example illustrates how the decorations and paraphernalia that were identified in the previous example can be referred to:

```
<text>
  ...
  <p>Een grote boom stond daar <deco decoRef="fig1" />,
  met aan zijn voet een mooie papaver <paraph paraphRef="obj1" />.</p>
  ...
  <p>Ik stel mij de pagina zo voor: <deco decoRef="fig2" /></p>
  ...
</text>
```

The **<decoration>** and **<paraphernalia>** elements and their unique DALF child elements are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT decoration %om.RR;
        (decoList | p+)>
<!ATTLIST decoration
        %a.global;>

<!ELEMENT decoList %om.RR;
        (decoItem+)>
<!ATTLIST decoList
        %a.global;>

<!ELEMENT decoItem %om.RR;
        (decoDesc, decoText?)>
<!ATTLIST decoItem
        %a.global;>

<!ELEMENT decoDesc %om.RR;
        (p+)>
<!ATTLIST decoDesc
        %a.global;>

<!ELEMENT decoText %om.RR;
        (p+)>
```

```
<!ATTLIST decoText
        %a.global;>

<!ELEMENT paraphernalia %om.RR;
        (paraphList | p+)>
<!ATTLIST paraphernalia
        %a.global;>

<!ELEMENT paraphList %om.RR;
        (paraphItem+)>
<!ATTLIST paraphList
        %a.global;>

<!ELEMENT paraphItem %om.RR;
        (paraphDesc, paraphText?)>
<!ATTLIST paraphItem
        %a.global;>

<!ELEMENT paraphDesc %om.RR;
        (p+)>
<!ATTLIST paraphDesc
        %a.global;>

<!ELEMENT paraphText %om.RR;
        (p+)>
<!ATTLIST paraphText
        %a.global;>
```

4.3.7 The physical condition: <condition>

The physical condition of the letter can be encoded inside one or more **<p>** elements in the **<condition>** element. There are no other than the global attributes for this element (3.3 Global attributes (p. 8)).

<p>
Contains the description of the physical condition of the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

The following example illustrates how the **<condition>** element can be used to record that a letter has the right corner missing, and has places on the folding lines that are torn:

```
<condition>
  <p>topright corner is missing</p>
  <p>the folding lines have caused some tears</p>
</condition>
```

The **<condition>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT condition %om.RR;
        (p+) >
<!ATTLIST condition
        %a.global;>
```

4.4 Presence of an envelope: <envOcc />

The envelope can contain valuable information for the contextualisation of a letter, or even contain text that may be closely related to the contents of the letter. However, the question whether or not to regard this text as part of the letter is a theoretical one. Some encoders may wish to exclude envelope contents completely; others may consider it relevant enough to include it as part of the letter. The DALF encoding scheme allows (and strongly suggests) to take a middle road, by providing the special **<envelope>** body element (see 5.1 The envelope: **<envelope>** (p. 47)) that enables the encoding of envelope content while at the same time keeping it separate from the letter proper.

In order to facilitate retrieval of documents in the DALF textbase, the element **<envOcc />** that explicates the presence or absence of an envelope is adopted as the fourth mandatory element of the letter description. It is an empty element that has one required attribute, **occ** (apart from the global attributes as defined in 3.3 Global attributes (p. 8)). When a letter has an envelope, this can be stated with value ‘yes’; when there is no envelope, the value ‘no’ should be used.

occ
Indicates the occurrence of an envelope. This is a mandatory attribute, with a required choice of ‘yes’ (the letter is accompanied by an envelope) or ‘no’ (the letter is not accompanied by an envelope).

Note that the **<envOcc />** element reflects some of the theoretical underpinnings of the **<envelope>** element. As will be explained later (see 5.1 The envelope: **<envelope>** (p. 47)), a functional definition of an envelope is proposed. This means that the part of a sending with addressing information is suggested to be encoded as an envelope. As the **<envOcc />** element in the header reflects the occurrence of the envelope and thus of the **<envelope>** element in the text body, this means that for example postcards should also have the value ‘yes’ for the attribute **occ**.

This example illustrates the encoding of a letter that has an envelope:

```
<envOcc occ="yes" />
```

The **<envOcc />** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT envOcc %om.RR;
        EMPTY>
```

```
<!ATTLIST envOcc
        %a.global;
        occ (yes | no) #REQUIRED>
```

4.5 The contents of the letter: <letContents>

The contents of the letter are of course a major source of interest for users of encoded DALF materials. In order to ensure a rich possibility of exploiting the DALF textbase, structured access to descriptive records of its cumulative contents is an interesting starting point. It can be used to implement a search mechanism that lets users do thematic queries throughout several (selections of) DALF letters, or provide the means to generate thematic editions like a regest or calendar edition. Therefore, the DALF header provides the possibility to summarise the contents of each separate letter within the **<letContents>** element. It is the first optional element that can appear after the four mandatory elements of **<letDesc>**.

In the description of the intellectual contents of a letter the encoder can specify systematically what the status of the intellectual content of the letter is regarding its completeness. That can be done by means of the following special attribute (in addition to the global attributes as specified in 3.3 Global attributes (p. 8)):

defective
Indicates whether the contents of the letter being described is defective, i.e. incomplete. This attribute is optional. Possible values are ‘yes’ for defective letters; ‘no’ for letters that are complete; or ‘unk’ when no statement can be made regarding the completeness of the intellectual content.

The description of the intellectual contents of a letter must consist of one or more paragraphs. This content description can be preceded by one or more **<class>** elements. That element is meant for a formal characterisation of the contents, according to a certain typology. No such typology exists as yet, however; we think of a functional-communicative "genre"-indication of letters which still has to be developed. In the meantime, the **<class>** element is specified as optional.

<class>
Contains a functional-communicative "genre"-indication of letters (still to be developed). Apart from global attributes, the following attribute is provided:

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute of value type CDATA.

<p>
Contains the description of the contents of the letter. This is a standard TEI

element (see [the TEI P4 Guidelines]).

<note>
Contains additional remarks, not yet covered in the preceding elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

As already mentioned, the **<class>** element is optional. Yet, when it appears, it has to occur before the first paragraph. At least one **<p>** element is required.

The following example contains a possible description of the contents of a letter in which Stijn Streuvels proposes to his girlfriend:

```
<letContents>
  <class>love letter</class>
  <p>Streuvels proposes to his girlfriend</p>
</letContents>
```

The **<letContents>** element and its unique DALF child element are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT letContents %om.RR;
        (class*, p+, note*)>
<!ATTLIST letContents
        %a.global;
        defective (yes|no|unk) "no">

<!ELEMENT class %om.RR;
        (#PCDATA)>
<!ATTLIST class
        %a.global;
        type CDATA #IMPLIED>
```

4.6 The history of the letter: <history>

Specific aspects of the history of a letter can be recorded in the **<history>** element, which is the second optional element inside **<letDesc>**. It is adopted quite literally from the Master encoding scheme for manuscript description records, with some adaptations to integrate it in the overall organisation of the DALF header. Although we had some reservations regarding the need for detailed historical meta-information for letters, we decided to provide this as an optional feature.

The historical description may cover details of the origination of the letter, the history of the letter before its acquisition by a holding institution, and the acquisition process by its holding institution.

The **<history>** element has no other than the global attributes (see 3.3 Global attributes (p. 8)), and contains a historical description in the following elements:

<origin>
Contains any descriptive or other information concerning the origin of a letter.

<provenance>
Contains any descriptive or other information concerning the history of a letter, after its creation but before its acquisition.

<acquisition>
Contains any descriptive or other information concerning the process by which a letter entered the holding institution.

<note>
Contains additional remarks, not yet covered in the preceding elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

None of these elements is required as such; yet at least one of **<origin>**, **<provenance>** or **<acquisition>** must be chosen when the **<history>** element is used. When they are used, the order specified must be respected. They all have the same structure. Only the global attributes (see 3.3 Global attributes (p. 8)) are applicable, and their respective descriptions have to be included in one or more paragraphs:

(Content model for **<origin>**, **<provenance>** and **<acquisition>**:)

<p>
Contains the description of a historical fact regarding the origination, respectively provenance or acquisition of the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

Note that the **<origin>** element is not meant to repeat the contents of the **<dateLet>** element inside **<letHeading>**, which already records when the letter was written (see 4.2.2 The place and time of writing: **<placeLet>** and **<dateLet>** (p. 23)). Instead, the **<origin>** element should be used to record additional facts about the origination of the letter, such as statements about draft versions. As for historical facts pertaining to the period after the acquisition of a letter by the holding institution, those fall outside the scope of **<provenance>** and **<acquisition>**. Instead they may be given in **<custodialHist>** inside **<adminInfo>** (cf. supra, 4.7.1 Administrative information: **<adminInfo>** (p. 39)).

The following example illustrates how an extensive historical description can be made for a letter of which a draft version was found dated 17 August 1924, and which has remained in the possession of the receiver until it was donated to the AMVC after his death, in 1937.

```
<history>
  <origin>
    <p>draft of letter found, dated 17 August 1924</p>
  </origin>
```

```
  <provenance>
    <p>between January 1925 and 3 September 1935, the letter resided
    in the private collection of the receiver</p>
    <p>between the death of the receiver on 3 September 1935 and 7
    February 1937, the letter was owned by the heirs of the reveiver</p>
  </provenance>
  <acquisition>
    <p>acquired by the AMVC through a gift by the heirs of the whole archive</p>
  </acquisition>
</history>
```

The **<history>** element and its unique DALF child elements are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT history %om.RR;
        ((origin, provenance?, acquisition?, note*) | (provenance, acquisition?, note*) |
(acquisition, note*))>
<!ATTLIST history
        %a.global;>

<!ELEMENT origin %om.RR;
        (p+)>
<!ATTLIST origin
        %a.global;>

<!ELEMENT provenance %om.RR;
        (p+)>
<!ATTLIST provenance
        %a.global;>

<!ELEMENT acquisition %om.RR;
        (p+)>
<!ATTLIST acquisition
        %a.global;>
```

4.7 Additional information: <additional>

Additional information about the letter is grouped in this third optional element of **<letDesc>**. Like the **<history>** element, it is adopted with only slight modifications from the Master encoding scheme. It enables the encoder to give a rich amount of additional information concerning the present custody and availability, other representations of the letter, accompanying materials, and further bibliographical information.

The **<additional>** element has only the global attributes (see 3.3 Global attributes (p. 8)), and can contain following elements:

<adminInfo>
Contains information about the present custody and availability of the letter.

<surrogates>
Contains information about any other formal representations that exist of the letter.

<accMat>
Contains details of any significant additional material which may be closely associated with the letter.

<listBibl>
Lists bibliographic descriptions of publications relating to the letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

<note>
Contains additional remarks, not yet covered in the preceding elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

None of these elements is required in particular; yet at least one of **<adminInfo>**, **<surrogates>**, **<accMat>** or **<listBibl>** must be present when **<additional>** is used. When some (or all) are used, they must occur in the order specified.

Because **<additional>** requires just the presence of one of its child elements without specifying any in particular, a minimal description of additional information for the letter in the following example could lack any of the child elements, provided that at least one remains and that the order in which they appear is respected:

```
<additional>
  <adminInfo>
    <availability>
      <p>Available under licence from the publishers.</p>
    </availability>
  </adminInfo>
  <surrogates>
    <p>a microfilm facsimile exists</p>
  </surrogates>
  <accMat>
    <p>half a page of the "De Gentenaar" journal of 16 January
    1932 has been included in the envelope</p>
  </accMat>
  <listBibl>
    <bibl>This letter is included in
      <author>Marcel De Smedt &amp;amp; Edward Vanhoutte</author>
      <title>Stijn Streuvels, De Teleurgang van den Waterhoek.
      Elektronisch-kritische editie/electronic-critical edition.</title>
      <pubPlace>Amsterdam</pubPlace>
      <publisher>Amsterdam University Press/KANTL</publisher>
      <date>2000</date>
      <idno type="ISBN">90-5356-441-1 (CD-Rom)</idno>
    </bibl>
  </listBibl>
</additional>
```

The **<additional>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT additional %om.RR;
        ((adminInfo, surrogates?, accMat?, listBibl?, note*) |
        (surrogates, accMat?, listBibl?, note*) |
        (accMat, listBibl?, note*) | (listBibl, note*))>
<!ATTLIST additional
        %a.global;>
```

4.7.1 Administrative information: <adminInfo>

This element is used to record statements about the availability of a letter, its custodial history and possible additional remarks regarding the cataloguing description the encoder would like to make.

The **<adminInfo>** element has only the global attributes (see 3.3 Global attributes (p. 8)) and can contain the following elements:

<availability>
Supplies information about the availability of a letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

<custodialHist>
Contains a description of the custodial history of a letter, inside one or more **<custEvent>** elements:

<custEvent>
Describes a single event related to the custodial history of a letter.

<note>
Contains additional remarks, not yet covered in the preceding elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

<remarks>
Contains any comments or remarks not forming part of the description proper, for use by the cataloguer.

<note>
Contains additional remarks, not yet covered in the preceding elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

None of these elements is required in particular; yet at least one of **<availability>**, **<custodialHist>** or **<remarks>** must be present. When some (or all) are used, they must occur in the order specified. The final descendants of each of these elements all have the same structure. Their respective descriptions have to be included in one or more paragraphs:

(Content model for **<availability>**, **<custEvent>** and **<remarks>**:)

<p>
Contains the description of a the availability status, respecively an event in the custodial history, or additional cataloguing remarks for a letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

All of the child elements of **<adminInfo>** can have the global attributes (see 3.3 Global attributes (p. 8)). Two of them have special attributes:

<availability>

status
Supplies a code indicating the current availability of a letter. This is an optional attribute with one of three predefined values. When a letter is freely available, this can be coded with the value ‘free’; when it is not, the value ‘restricted’ can be used; when the availability status is unknown, the value ‘unknown’ can be used.

<custEvent>

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

Because **<adminInfo>** requires just the presence of one of its child elements without specifying any in particular, a minimal description of additional administrative information for the letter in the following example could lack any of the child elements, provided that at least one remains and that the order in which they appear is respected:

```
<adminInfo>
  <availability status="restricted">
    <p>Available for academic research purposes only.</p>
  </availability>
  <custodialHist>
    <custEvent type="loan">
      <p>from 13 January 1955 to 3 March 1956, the letter was given on loan to
      the university of Amsterdam</p>
    </custEvent>
  </custodialHist>
  <remarks>
    <p>maybe this letter can turn out significant for the retracing of the
    missing letters by Streuvels to Excelsior between 1926 and 1928</p>
  </remarks>
</adminInfo>
```

The **<adminInfo>** element and its unique DALF child elements are declared in the

DALFExtns.dtd file as follows:

```
<!ELEMENT adminInfo %om.RR;
        ((availability, custodialHist?, remarks?, note*) |
        (custodialHist, remarks?, note*) | (remarks, note*))>
<!ATTLIST adminInfo
        %a.global;>

<!ELEMENT custodialHist %om.RR;
        (custEvent+, note*)>
<!ATTLIST custodialHist
        %a.global;>

<!ELEMENT custEvent %om.RR;
        (p+)>
<!ATTLIST custEvent
        %a.global;
        type CDATA #IMPLIED>

<!ELEMENT remarks %om.RR;
        (p+)>
<!ATTLIST remarks
        %a.global;>
```

4.7.2 Other representations: <surrogates>

If a letter has incarnations in other representation formats, those can be described in the optional **<surrogates>** element. Typically, this will be digital or photographic copies of the letter.

The **<surrogates>** element has only the global attributes (see 3.3 Global attributes (p. 8)), and contains a description of the representation in one or more paragraphs:

<p>
Contains the description of a representation in another form of a letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

The following example illustrates the description of a digital facsimile of a letter:

```
<surrogates>
  <p>
    <bibl>
      digital facsimile, saved as JPEG image file
      <title type="filename">LS081141.jpg</title>
      <idno>AMVC 911/5 jpg</idno>
      <date>September 1996</date>
    </bibl>
  </p>
</surrogates>
```

The **<surrogates>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT surrogates %om.RR;
        (p+)>
<!ATTLIST surrogates
        %a.global;>
```

4.7.3 Accompanying materials: <accMat>

Sometimes, some external materials can be very closely related to a letter. They can accompany a letter in the same envelope, or can be stored together in an archive for some reason. Such accompanying materials can be described with the optional **<accMat>** element.

The **<accMat>** element has one special attribute apart from the global ones (see 3.3 Global attributes (p. 8)):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

It provides a description for each piece of accompanying material in at leas one paragraph:

<p>
Contains the description of a piece of material which accompanies or is closely related to a letter. This is a standard TEI element (see [the TEI P4 Guidelines]).

The following example shows the description of a copied tax form which accompanied a letter:

```
<accMat type="external_document">
  <p>A copy of a tax form from 1947 is included in the envelope
  with the letter. It is not catalogued separately.</p>
</accMat>
```

The **<accMat>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT accMat %om.RR;
        (p+)>
<!ATTLIST accMat
        %a.global;
        type CDATA #IMPLIED>
```

4.7.4 Appearance in published materials: <listBibl>

The last optional element inside **<additional>** is **<listBibl>**. This is a standard TEI element that in this context should hold information about representations of the letter available within any published works. When for example a letter has been included in an existing edition, a bibliographical record for that edition can be provided inside one or more **<bibl>** or **<biblFull>** elements inside **<listBibl>** (for an exhaustive listing of the possible contents and attributes of **<listBibl>**, see [the TEI P4 Guidelines]). The purpose of this **<listBibl>** element is thus clearly distinct from that of the **<surrogates>** and **<accMat>** elements described earlier: **<surrogates>** (see 4.7.2 Other representations: **<surrogates>** (p. 41)) only describes other forms in which the letter is stored, and **<accMat>** describes materials accompanying the letter.

For a letter that has been included in a previously published edition, the following example illustrates how the **<listBibl>** element inside **<additional>** is the right place for a bibliographical description of this edition. Note that for illustrative purposes, two editions are described: an electronic and an 'analog' one:

```
<listBibl>
  <bibl>This letter is included in
    <author>Marcel De Smedt &amp;amp; Edward Vanhoutte</author>
    <title>Stijn Streuvels, De Teleurgang van den Waterhoek.
    Elektronisch-kritische editie/electronic-critical edition.</title>
    <pubPlace>Amsterdam</pubPlace>
    <publisher>Amsterdam University Press/KANTL</publisher>
    <date>2000</date>
    <idno type="ISBN">90-5356-441-1 (CD-Rom)</idno>
  </bibl>
  <bibl>This letter is included in
    <author></author>
    <title>Uitgeverij Lannoo, de vroege jaren</title>
    <pubPlace>Tielt</pubPlace>
    <publisher>Lannoo uitgeverij</publisher>
    <date>2001</date>
  </bibl>
</listBibl>
```

4.8 Distinct letter parts: <letPart>

Letters are typically rather short documents, written by one author at one place, most often in one and the same time span and forming one structural whole (possibly over a couple of pages). Yet, sometimes the encoder may wish to distinguish several distinct subunits in one letter, and describe the specific characteristics of each part separately. This can be done within several instances of the **<letPart>** element, which is the last optional element in **<letDesc>**.

The **<letPart>** element has only the global attributes (see 3.3 Global attributes (p. 8)). Its inner structure shows heavy resemblance to that of the **<letDesc>**

element as a whole, because it should allow the encoder to describe the specific particularities of the letter subdivisions in their own right. However, there are some important differences. In contrast to **<letDesc>**, the **<letPart>** element does not contain a **<letIdentifier>** element, as this can only be used to give a cataloguing description of the *whole* letter as a physical entity. Instead, it provides an **<idno>** element to identify the letter part within the letter itself. Another missing element in **<letPart>** is the **<envOcc />** element, since also the statement about the occurrence of an envelope can only pertain to the letter as a whole. The following elements can occur within **<letPart>**:

<idno>
Contains an identification code for a letter part. This is a standard TEI element (see [the TEI P4 Guidelines]).

<letHeading>
Contains a structured description of specific bibliographical information of a letter part. This is a unique DALF element, described at 4.2 The letter heading: **<letHeading>** (p. 18) .

<physDesc>
Contains a description of the specific physical appearance of a letter part. This is a unique DALF element, described at 4.3 The physical description: **<physDesc>** (p. 24) .

<letContents>
Contains a description of the specific intellectual contents of the letter part. This is a unique DALF element, described at 4.5 The contents of the letter: **<letContents>** (p. 34) .

<history>
Contains a description of the specific history of the letter part. This is a unique DALF element, described at 4.6 The history of the letter: **<history>** (p. 35) .

<additional>
Groups specific additional information about the letter part. This is a unique DALF element, described at 4.7 Additional information: **<additional>** (p. 37) .

<letPart>
Contains metadata about distinct parts inside a letter part. This is a unique DALF element, described at 4.8 Distinct letter parts: **<letPart>** (p. 43) .

<note>
Contains additional information about the letter that is not covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

Note that in contrast to the **<letDesc>** element, **<letPart>** does not require any particular child element; it does require, however, that at least one of **<idno>**, **<letHeading>**, **<physDesc>**, **<letContents>**, **<history>** or **<additional>** be present.

After all, the fact that an encoder distinguishes a distinct part in a letter implies that it should have at least some specific characteristics. Furthermore, the elements that do occur, must do so in the order specified. The encoder should recall that specific characteristics of separate letter subdivisions may only be provided *after* the required **<letDesc>** element descriptions for the letter as a whole, and that the elements occurring inside **<letPart>** elements have the same structural requirements as when used directly under **<letDesc>**.

The following example shows how different **<letPart>** elements can capture the particular specific characteristics of a response letter written by Stijn Streuvels letter that is written on the verso side of an original letter written to him by Joris Lannoo:

```
<letDesc>
  <letIdentifier>
    <country>Belgium</country>
    <settlement>Antwerp</settlement>
    <repository>AMVC</repository>
    <collection>S 154/1235</collection>
    <idno>223949</idno>
  </letIdentifier>
  <letHeading>
    <author>Stijn Streuvels</author>
    <addressee>Lannoo</addressee>
    <placeLet>Ingooigem</placeLet>
    <dateLet>1943-04-24</dateLet>
  </letHeading>
  <physDesc>
    <type>letter</type>
    <support>
      <p>single page with pre-printed letterhead, writing on both sides</p>
    </support>
    <extent>
      <dimensions>
        <height units="mm">214</height>
        <width units="mm">276</width>
      </dimensions>
    </extent>
  </physDesc>
  <envOcc occ="yes" />
  <letPart>
    <letHeading>
      <author>Joris Lannoo</author>
      <addressee>Stijn Streuvels</addressee>
      <placeLet>Tielt</placeLet>
      <dateLet>1943-04-20</dateLet>
    </letHeading>
    <physDesc>
      <type>letter</type>
      <support>
        <p>typed letter on recto side of the letter</p>
      </support>
      <extent>
        <dimensions>
```

```
            <height units="mm">214</height>
            <width units="mm">276</width>
          </dimensions>
        </extent>
      </physDesc>
      <letContents>
        <p>Joris Lannoo asks Streuvels what to do ...</p>
      </letContents>
    </letPart>
    <letPart>
      <letHeading>
        <author>Joris Lannoo</author>
        <addressee>Stijn Streuvels</addressee>
        <placeLet>Tielt</placeLet>
        <dateLet>1943-04-20</dateLet>
      </letHeading>
      <physDesc>
        <type>letter</type>
        <support>
          <p>written letter on verso side of the letter</p>
        </support>
        <extent>
          <dimensions>
            <height units="mm">214</height>
            <width units="mm">276</width>
          </dimensions>
        </extent>
      </physDesc>
      <letContents>
        <p>Streuvels advises Lannoo to ...</p>
      </letContents>
    </letPart>
</letDesc>
```

The **<letPart>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT letPart %om.RR;
        ((idno, letHeading?, physDesc?, letContents?, history?, additional?, letPart*, note*) |
        (letHeading, physDesc?, letContents?, history?, additional?, letPart*, note*) |
        (physDesc, letContents?, history?, additional?, letPart*, note*) |
        (letContents, history?, additional?, letPart*, note*) |
        (history, additional?, letPart*, note*) |
        (additional, letPart*, note*) |
        (letPart+, note*))>
<!ATTLIST letPart
        %a.global;>
```

5. Letter-specific textual features

A searchable electronic textbase of correspondence material not only requires specific provisions for the encoding of letter-specific metadata. Also the transcription proper of the letters calls for means to encode letter-specific features. Although many of the features encountered in letters can be covered with standard TEI elements (in particular, those described for the transcription of primary sources in chapter [18. Transcription of Primary Sources] of the TEI P4 Guidelines), there are some very specific ones for which the semantics of TEI elements would have to be stretched to an intolerable degree. Obviously, there are some structural elements that are unique to letters, like the envelope and postscripts. Others are more generally bound to primary manuscript materials, and thus occur very frequently in letters, such as calculations, pre- and post-printed materials, and decorative elements. One element which is not specific for manuscript material, but a refinement of the TEI provisions to mark dimensions, is taken over from the Master DTD.

The next sections will explain and motivate the DALF elements introduced to cover these features, and illustrate their us(ag)e.

5.1 The envelope: <envelope>

The typical letter is delivered within an envelope. Often, when letters are stored, their accompanying envelopes are stored with them. For an encoder, there are good reasons to provide transcriptions of letters with a transcription of their envelopes. One is that envelopes may contain valuable information for the identification of letters. When the letter is lacking some of the indicators of the communicative context that are important for an unambiguous identification of a letter (communicative participants, time and place of writing; see 4.2 The letter heading: **<letHeading>** (p. 18)), there is good chance they still can be deduced from the postal information on the envelope. Furthermore, the envelope may contain significant information, apart from the postal data. Some authors create on their envelopes pieces of art in their own right, that may closely relate and contribute to the letter content. Some receivers may use the envelope to write quick notes about the letter content or other contextual circumstances. Envelopes may even contain drafts of successive letters.

From a technical point of view, envelopes have certain typical formal and semantic features that justify the adoption of an own tag, e.g. the occurrence of a postmark, addresses on front and/or back, possibly additional plain text and/or graphics, or no text at all, and even the containment of another envelope. Therefore, the **<envelope>** tag is introduced as a direct child of the TEI **<text>** element, at the same structural level as the TEI elements **<front>**, **<body>** and **<back>**. It may occur anywhere before, in between or after those.

<envelope>
Contains the information on the envelope.

However desirable the adoption of a special **<envelope>** element may be, care has to be taken with regards to its semantics. After all, what holds for a letter, also holds for the envelope of a letter: they come in many forms and flavours, pushing the encoder to the question of what *is* an envelope. Is it the separate piece of paper that holds the postal information of a typical letter? Should the box containing a parcel then be regarded as an envelope, too? What to do with postcards, that most often are not packaged in a separate envelope, but still manage to reach the right destination? Perhaps the best option is not to take a strictly physical criterium, but instead abstract this into a functional view, thus defining an envelope as 'the part/room/space *reserved for* postal information'. This definition includes the postal information, as well as all other possible contents of that functional part of the letter. In order to specify the kind or format of envelope concerned, the **<envelope>** element has one special attribute apart from the global ones (see 3.3 Global attributes (p. 8)):

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA. For example, the address part of a postcard can be encoded as an envelope of **type** 'postcard'.

Note that the functional approach to the encoding of an envelope can be illustrated with the 'envelope part' of a postcard, which is less physically distinct from that of a normal letter:

- When a postcard has no addressing information, but instead has text of the letter running on on the part normally reserved for addressing information, the envelope can be encoded as empty.
- When it has no addressing information, but has some text that is clearly distinct from the letter on the part normally reserved for addressing information, this can be encoded as text on the envelope.

The envelope can contain following elements:

<envPart>
Contains the information on one side of the envelope.

<envelope>
Contains the information on an enclosed envelope.

<note>
Contains additional remarks the encoder would wish to make. This is a standard TEI element (see [the TEI P4 Guidelines]).

These elements may occur in any order and frequency, provided that at least one of them is present. The recursive **<envelope>** element enables the encoding of envelopes that are enclosed within another envelope. Note that the **<note>** element may well be the only element of an envelope (in contrast to the *additional* **<note>** elements in the header elements described thus far): this allows to explicate the encoding of an envelope that has no informational contents.

The **<envPart>** element contains all data that appear on one side of the envelope. To indicate the part concerned, one special attribute is declared, apart from the global ones (see 3.3 Global attributes (p. 8)):

side
Describes which side of the envelope features the data described. It is an optional attribute with one of three predefined values. The encoder can indicate whether the data appears at the front side of the envelope with value ‘front’; on the back side with value ‘back’; or on a one-sided postcard envelope with value ‘postcard’.

An envelope part can hold postal information like addresses of sender and receiver and a postmark that may be important for the identification of the letter. Also, additional textual information can be encoded within an envelope part. The following elements are used:

<address>
Contains a postal or other address, for example of a publisher, an organisation, or an individual. This is a standard TEI element (see [the TEI P4 Guidelines]), with one added attribute, however:

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA. Sensible values include ‘receiver’ for the address of the receiver, and ‘sender’ for that of the sender.

<postmark>
Contains the postmark on an envelope.

** **
Contains a subdivision of the envelope; non-postal information on the envelope should be encoded within this element. This is a standard TEI element (see [the TEI P4 Guidelines]).

elements of the Incl class
Elements which may appear at any point within a TEI text. This is a standard TEI element class, with some specific DALF modifications (see 7. Modifications to TEI element classes (p. 69)).

These elements may occur in any order and frequency, provided that at least one is present. .

The **<postmark>** element contains a transcription of the postmark on the letter. This may be useful for the identification of the date and place of writing, when they are not stated anywhere else. Also, graphical aspects of the postmark can be described. This can be done with the following elements:

<figure>
Indicates the location of a graphic, illustration, or figure. This is a standard TEI element (see [the TEI P4 Guidelines]).

<placeName>
Contains an absolute or relative place name. This is a standard TEI element (see [the TEI P4 Guidelines]).

<date>
Contains a date in any format. This is a standard TEI element (see [the TEI P4 Guidelines] .

<note>
Contains additional remarks, not yet covered by any other of the previous elements. This is a standard TEI element (see [the TEI P4 Guidelines]).

These elements may occur in any order and frequency, provided that at least one is present. The **<note>** element may only be used when at least one of the other elements is present.

The following example illustrates how an envelope can be encoded that contains an address of the receiver, a postmark and the sentence "Dank u, postbode!" on the front side, and the address of the sender on the back side. Furthermore, it holds a postcard that has no postal or other information on its address part.

```
<envelope type="220x110">
  <envPart side="front">
    <address type="receiver">
      <addrLine>Stijn Streuvels</addrLine>
      <addrLine>Lijsternest</addrLine>
      <addrLine>Ingooigem</addrLine>
    </address>
    <postmark>
      <placeName>Tielt</placeName>
      <date value="1924-10-24">24.10.'24</date>
    </postmark>
    <div>
      <p>Dank u, postbode!</p>
    </div>
  </envPart>
  <envPart side="back">
    <address type="sender">
      <addrLine>Lannoo uitgeverij</addrLine>
      <addrLine>Meulebeekschesteenweg 641</addrLine>
```

```
      <addrLine>Tielt</addrLine>
    </address>
  </envPart>
  <envelope type="postcard">
    <note>this envelope contains no data</note>
  </envelope>
</envelope>
```

The following lines demonstrate how the DALFExtns.dtd file redefines the **<text>** element to include **<envelope>**, and how the latter element itself and its unique DALF child elements are defined:

```
<!ELEMENT text %om.RR;
        ((envelope | %m.Incl;)*,
        (front, (envelope | %m.Incl;)*)?,
        (body | group), (envelope | %m.Incl;)*,
        (back, (envelope | %m.Incl;)*)?)>
<!ATTLIST text
        %a.global;
        %a.declaring;>

<!ELEMENT envelope %om.RR;
        (envPart | envelope | note)+>
<!ATTLIST envelope
        %a.global;
        type CDATA #IMPLIED>

<!ELEMENT envPart %om.RR;
        ((address | postmark | div | %m.Incl;)+)>
<!ATTLIST envPart
        %a.global;
        side (front | back | postcard) #IMPLIED>

<!ELEMENT address %om.RR;
        ((%m.Incl;)*, ((addrLine, (%m.Incl;)*)+ | ((%m.addrPart;), (%m.Incl;)*)* ))>
<!ATTLIST address
        %a.global;
        type CDATA #IMPLIED>

<!ELEMENT postmark %om.RR;
        ((figure | placeName |date)+, note*)>
<!ATTLIST postmark
        %a.global;>
```

5.2 The postscript: <ps>

Postscripts are a typical phenomenon for letters. Occurring after the closing formulae and salutation, they form a last addition to the contents of the letter. Moreover, the author often explicitly signals this additional status with the abbreviation ‘P.S.’.

Their formulaic use and meaning justify an own tag. Therefore, the **<ps>** element is adopted in the DALF DTD. Because it can appear only at the end of letters, or letter parts written by separate authors, it is specified as a member of the TEI element class **divbot** (see [the TEI P4 Guidelines]).

<ps>
Contains a postscript in a letter or letter part.

The **<ps>** element has only the global attributes. It can contain all elements from the TEI **specialPara** element class (see [the TEI P4 Guidelines]). This is the same content model as for example the **<note>** and **<add>** elements, which should be broad enough to cater for all elements that can occur within postscripts.

The following example shows the closing part of a letter containing a postscript:

```
<closer>
  <salute>Met vriendelijken groet</salute>
  <signed>(Styn Streuvels)</signed>
</closer>
<ps>
  <p id="xr2"><add id="add1"><abbr expan="postscriptum">P.S.</abbr> Ze jubileeren
  bij de firma Veen (60 jaar bestaan)<ref target="n8">8</ref> en er wordt me daarom
  gevraagd, door het comit&eacute;: hoeveel geld ik daarvoor als feestgave wensch te
  geven! Zonderlinge zeden? Als ik nu eens vroeg: hoeveel ze voor mij beschikken als
  75-jarige jubilaris!</add></p>
</ps>
```

The **<ps>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT ps %om.RR;
        %specialPara;>
<!ATTLIST ps
        %a.global;>
```

5.3 Calculations: <calc>

Modern correspondence, especially businesslike letters, may contain a lot of numerical data. The contents of such calculations may be very valuable for research. Formally, calculations are often set apart from running text, and it may be desirable to mark them with explicit encoding features. This provides researchers with greater control over the textual features they want to study.

Calculations have an internal structure the semantics of which cannot be captured sufficiently with the standard TEI **<num>** element (see [the TEI P4 Guidelines]). We considered the option to incorporate MathMl into the DALF DTD, which is an existing W3C standard providing a specialised tagset for mathematical

formulae (see [the TEI P4 Guidelines]). In chapter 22.2 of the TEI P4 guidelines directions are given for specifying external tagsets as XML notations that can be used in the **<formula>** tag. However, testing that mechanism with the literal examples given in that chapter turned out unsuccessful. Further investigation of postings on the TEI public mailing list (see [the TEI P4 Guidelines]) showed that other TEI users encountered the same problem, and learned that the incorporation mechanism itself does not provide the inclusion functionality we had in mind. These troubles and the unwieldy suggestions to get around the incorporation of external tagsets like MathMl [3], and the complexity of the MathMl standard itself made this option less favourable than devising a specialised element that can encode at least some of the semantic structure of calculations in a rudimentary way.

This specialised element for calculations adopted in the DALF DTD is **<calc>**. Because calculations may occur on virtually any physical place on a letter or envelope, the **<calc>** element is declared as member of the **common**, **inter** and **tpParts** element classes [4], defined respectively at [the TEI P4 Guidelines] , [the TEI P4 Guidelines] and [the TEI P4 Guidelines] .

> **<calc>**
> Contains a calculation.

The **<calc>** element has only the global attributes (see 3.3 Global attributes (p. 8)). Calculations may partly or entirely consist of plain prose (thus possibly needing some phrase-level TEI elements), and can contain embedded calculations. The basic structure, however, is made up of one or more arguments, an operator, and a result. This structure is captured in following elements:

> **elements of the phrase class**
> Groups PCDATA and those elements which can occur at the level of individual words or phrases. This is a standard TEI element class (see [the TEI P4 Guidelines]).
>
> **<calc>**
> Contains a calculation.
>
> **<arg>**

.....

[3] A full technical discussion can be found in Arjan Loeffers answer on 28 October 1997 to Andreas Nolda who formulated nearly the same problem we encountered. These messages can be found in the TEI List archive.

.....

[4] This distribution is based on that of the original TEI <**figure**> element. Note that there is a slight discrepancy between the TEI documentation of this element (see [the TEI P4 Guidelines]) and its declaration in the DTD files. The documentation states that <**figure**> is a member of only 2 element classes, namely **inter** and **tpParts**; whereas the TEI teiclas2.ent file declares <**figure**> as member of a third element class: **common**. Since in practice the distribution of <**figure**> clearly suggested the latter evidence, the DALF scheme also defines all elements whose distribution is modelled after that of the original TEI <**figure**> element as members of the **common**, **inter**, and **tpParts** element classes.

Contains the argument of a calculation.

<oper>
Contains the operator of a calculation.

<result>
Contains the result of a calculation.

These elements may occur in any order and frequency. Arguments and results have a similar inner structure: they can contain plain prose text and possibly phrase-level TEI elements, or another calculation, and possibly some more or less free-standing arguments. Operators are predominantly in PCDATA form, but allow phrase-level TEI elements to appear within.

contents of **<arg>** and **<result>**:

elements of the phrase class
Groups PCDATA and those elements which can occur at the level of individual words or phrases. This is a standard TEI element class (see [the TEI P4 Guidelines]).

<calc>
Contains an (embedded) calculation.

<arg>
Contains an (embedded) argument.

contents of **<oper>**:

elements of the phrase class
Groups PCDATA and those elements which can occur at the level of individual words or phrases. This is a standard TEI element class (see [the TEI P4 Guidelines]).

All of these elements may contain any combination of child elements.

The following example shows how a calculation in the original can be encoded with the **<calc>** element. Note the occurrence of a mixture of PCDATA and phrase-level elements within the **<calc>** (sub)structure(s), and the encoding of the embedded **<calc>**, as a child of the second argument:

```
<calc>
  <arg>969 <abbr expan="exemplaren">ex.</abbr> (zie afrekening van 30.8.41)</arg>
  <oper>-</oper>
  <arg>138<abbr expan="exemplaren">ex</abbr> (
    <calc>
      <arg>133 <abbr expan="exemplaren">ex.</abbr> verkocht</arg>
      <oper>+</oper>
      <arg>5<abbr expan="persexemplaren">persex.</abbr></arg>
    </calc>)
```

```
  </arg>
  <result><hi rend="double_underlined">831</hi><abbr
expan="exemplaren">ex.</abbr></result>
</calc>
```

The **<calc>** element and its unique DALF child elements are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT calc %om.RR;
        (%phrase; | calc | arg | oper | result)*>
<!ATTLIST calc
        %a.global;>

<!ELEMENT arg %om.RR;
        (%phrase; | calc | arg)*>
<!ATTLIST arg
        %a.global;>

<!ELEMENT oper %om.RR;
        (%phrase;)*>
<!ATTLIST oper
        %a.global;>

<!ELEMENT result %om.RR;
        (%phrase; | calc | arg)*>
<!ATTLIST result
        %a.global;>
```

5.4 Pre- and post-printed material: <print>

Letters may be written (or printed) on paper (or other support material) containing pre-printed text like letterheads, form data, newspaper articles, ads and so on. There are also similar formal text elements, like stamps, that may be added after the composition of the letter. Such text fragments can be seen as part of the letter, but may need to be distinguished from more "authorial" parts of the letter, as they mostly have an impersonal character. It is imaginable that such material would be excluded from e.g. a linguistic study on the language of a writer, or selected in a study on stamps in letters.

The TEI tagset does not contain any element that can accurately indicate pre-printed text material. Post-printed material, like stamps, could possibly be tagged with the TEI **<add>** element. However, as that element is reserved for ‘letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector’, it is questionable whether mostly impersonal stamps can be regarded as genuine additions in that sense. Therefore, in order to provide consistent treatment for all pre- and post-printed material, a special element is presented in the DALF DTD: **<print>**. Since pre- and post-printed material may occur on virtually any physical place in a

letter or on an envelope, the **<print>** element is declared as member of the **Incl** element class, as described at 7. Modifications to TEI element classes (p. 69) .

<print>
Contains printed material that was present on the carrier of the letter, or added afterwards.

The **<print>** element has a number of attributes, apart from the global ones (see 3.3 Global attributes (p. 8)), that are borrowed from some TEI elements. It has the same attributes **style**, **character** and **ink** as the TEI **<handShift />** element, to indicate particularities regarding the writing style, the font and ink used. Further, the **hand** attribute is copied from the TEI element **<add>**, to enable reference to a hand the encoder has defined earlier in the header under **<hand>**. The last attribute, **type**, enables the characterisation of the type of printed material, like ‘letterhead’, ‘form’, ‘newspaper’, ‘ad’, ‘stamp’ and so on:

Borrowed from the TEI element **<handShift />** (see [the TEI P4 Guidelines]):

style
Indicates recognized writing styles, like e.g. ‘secretary’, ‘copperplate’, ‘Chancery’, ‘Italian’, etc. It is an optional attribute, with value type CDATA.

ink
Describes tint or type of ink, e.g. ‘brown’. May also be used to indicate the writing medium, e.g. ‘pencil’. This is an optional attribute, with CDATA as its value type.

character
Describes other characteristics of the hand, particularly those related to the quality of the writing, such as ‘shaky’, ‘thick’, ‘regular’. It is an optional attribute, of value type CDATA.

Borrowed from the TEI element **<add>** (see [the TEI P4 Guidelines]):

hand
Signifies the hand of the agent which made the addition. This is an optional attribute, with IDREF as value type. Its value must thus be one of the hand identifiers declared in the document header.

Other:

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

The **<print>** element has no specific inner structure: it is used to encode plain PCDATA text. In order to allow normal textual subelements, the same content model

as for the TEI elements **<note>** and **<add>** is used, viz. **specialPara** (see [the TEI P4 Guidelines]).

The following example illustrates the encoding of a printed letterhead:

```
<print type="letterhead">FRANK&middot;LATEUR</print>
```

The **<print>** element is declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT print %om.RR;
        %specialPara;>

<!ATTLIST print
        %a.global;
        style CDATA #IMPLIED
        ink CDATA #IMPLIED
        character CDATA #IMPLIED
        hand IDREF #IMPLIED
        type CDATA #IMPLIED>
```

5.5 Decorative elements: <deco /> and <paraph />

Letters can contain all sorts of material which is strictly speaking non-textual. Authors can include illustrative or plainly decorative sketches in their letters, or even include real objects like dried flowers, hair curls etc.

The TEI scheme provides the possibility to mark decorative elements with the **<figure>** element (see [the TEI P4 Guidelines]). However, in view of the archival requirements for DALF letters, we opted for a more comprehensive approach to mark up these text elements. As was explained in a previous section (4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)), the DALF header contains a **<decoration>** and **<paraphernalia>** element where the different decorative elements in the letter can be described and identified. Where those elements are encountered in the text, the encoder can indicate this and refer back to those definitions by means of the special DALF elements **<deco />** and **<paraph />**. These differentiate between decorations, i.e. less-textual materials that can be regarded to have originated within the writing act; and paraphernalia, materials that did not originate within the writing act. Decorations can include drawings, schemes etc., whereas paraphernalia can include hair curls, dried flowers etc. Since these phenomena can occur very freely within their textual context, they have been defined as members of the TEI **Incl** class (see 7. Modifications to TEI element classes (p. 69)).

<deco />
Marks the occurrence of a decoration, i.e. a decorative element that has originated within the writing act.

<paraph />
Marks the occurrence of a parapharnalia, i.e. a decorative element that has originated independently of the writing act.

They are empty elements that merely indicate the occurrence of decorative elements in the text. They must refer to the definition of the element concerned, and may also include a reference to a digitally scanned facsimile of the decorative item. Therefore, the following attributes are specified, apart from the global ones (see 3.3 Global attributes (p. 8)):

decoRef (for <deco />), resp. paraphRef (for <paraph />)
Provides a pointer to the definition of the decorative element in the header. This is a required attribute, with IDREF as value type. Its value must thus be the name of an identifier of a **<decoItem>** element the header (see 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)).

entity
Provides a pointer to an external entity within which the graphic image of the decoration is stored. It is an optional attribute with value type ENTITY. Its value must thus be the name of an entity that is declared at the beginning of the electronic document (see for detailed description [the TEI P4 Guidelines]).

Note that it is up to the encoder to decide whether to encode decorative material that was already present on the paper before the letter was written or added afterwards, as a decorative element (**<paraph />**) or as pre/post-printed material (**<print>**). An important factor for this choice, however, is the function of the decorative material within the text. If the author of the letter clearly intended some relation between his writing and such a decorative element, it could be considered a real decoration (possibly within a **<print>** element). If on the other hand there does not seem to be a real relation, it can just be tagged as printed material without further specifications.

The following example illustrates how the occurrence of two decorations and one paraphernalia can be indicated and linked both to their definition in the header (that is assumed to have been provided in the manner specified in 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)) as well as to their visual representations:

```
<?xml version="1.0"?>
<!DOCTYPE TEI.2 PUBLIC "//CTB//DTD Dalf 1.0 (based on TEI)//NL"
"DALF.dtd" [
<!NOTATION jpeg  PUBLIC
   'ISO DIS 10918//NOTATION JPEG Graphics Format//EN' >
<!NOTATION png PUBLIC
   '-//TEI//NOTATION IETF RFC2083 Portable Network Graphics//EN'>
<!ENTITY fig1 SYSTEM "fig1.jpg" NDATA jpeg>
<!ENTITY fig2 SYSTEM "fig2.jpg" NDATA jpeg>
<!ENTITY obj1 SYSTEM "obj1.png" NDATA png>
]>
```

```
<TEI.2>
  <text>
    ...
    <p>Een grote boom stond daar <deco decoRef="fig1" entity="fig1" />,
    met aan zijn voet een mooie papaver <paraph paraphRef="obj1" entity="obj1" />.</p>
    ...
    <p>Ik stel mij de pagina zo voor: <deco decoRef="fig2" entity="fig2" /></p>
    ...
  </text>
</TEI.2>
```

The **<deco />** and **<paraph />** elements are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT deco %om.RR;
        EMPTY>
<!ATTLIST deco
        %a.global;
        decoRef IDREF #REQUIRED
        entity ENTITY #IMPLIED>

<!ELEMENT paraph %om.RR;
        EMPTY>
<!ATTLIST paraph
        %a.global;
        paraphRef IDREF #REQUIRED
        entity ENTITY #IMPLIED>
```

5.6 Detailed dimensions: <dimensions>

The **<dimensions>** element is a specialised means to express the size of 3-dimensional objects taken over from the Master DTD (with slight adaptations). Since descriptions of dimensions can occur on various places inside normal running text, it is declared as member of the TEI class **phrase** (see [the TEI P4 Guidelines]).

<dimensions>
Contains any kind of dimensional specification.

The dimensions expressed in the **<dimensions>** element can be specified with some special attributes, apart from the global ones (see 3.3 Global attributes (p. 8)):

units
Specifies the units used for this measurement. This is an optional element with CDATA as value type. Sensible values are abbreviations for measurement units like ‘cm’, ‘mm’, ‘inch’.

type

Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

The **<dimensions>** element can contain PCDATA and necessary phrase-level elements, or can contain specific elements that identify the length, width or depth:

elements of the phrase class
Groups PCDATA and those elements which can occur at the level of individual words or phrases. This is a standard TEI element class (see [the TEI P4 Guidelines]).

<height>
Contains a measurement for the height of an object.

<depth>
Contains a measurement for the depth of an object.

<width>
Contains a measurement for the width of an object.

These elements may occur freely in any combination and frequency. They all have a special attribute (apart from the global ones defined at 3.3 Global attributes (p. 8)) to express the measurement unit used:

units
Specifies the units used for this measurement. This is an optional element with CDATA as value type. Sensible values are abbreviations for measurement units like ‘cm’, ‘mm’, ‘inch’.

They can contain PCDATA and necessary phrase-level elements as their contents:

elements of the phrase class
Groups PCDATA and those elements which can occur at the level of individual words or phrases. This is a standard TEI element class (see [the TEI P4 Guidelines]).

The following example illustrates how the description of the dimensions of a cover in a letter can be expressed with the **<dimensions>** element:

```
... het kaft is <dimensions>
   <height units="cm">40 cm</height>
   hoog en
   <width units="cm">25 cm</width>
   breed
</dimensions>
```

The **<dimensions>** element and its unique child elements are declared in the DALFExtns.dtd file as follows:

```
<!ELEMENT dimensions %om.RR;
```

```
        (%phrase;|height|width|depth)*>
<!ATTLIST dimensions
        %a.global;
        %a.measured;
        type CDATA #IMPLIED>

<!ELEMENT height %om.RR;
        (%phrase;)*>
<!ATTLIST height
        %a.global;
        %a.measured;>

 <!ELEMENT depth %om.RR;
         (%phrase;)*>
 <!ATTLIST depth
         %a.global;
         %a.measured;>

<!ELEMENT width %om.RR;
        (%phrase;)*>
<!ATTLIST width
        %a.global;
        %a.measured;>
```

6. Correlations of logical and physical structures

The encoding of letters, and primary manuscript materials in general, can be problematic for XML-based encoding schemes like TEI and DALF. The problem arises out of one of the premises of XML, namely the conception of a document as an Ordered Hierarchy of Content Objects (OHCO). This requires documents to take on a properly nesting hierarchy of container elements. This means that elements can only contain true subsets of other elements, like for example:

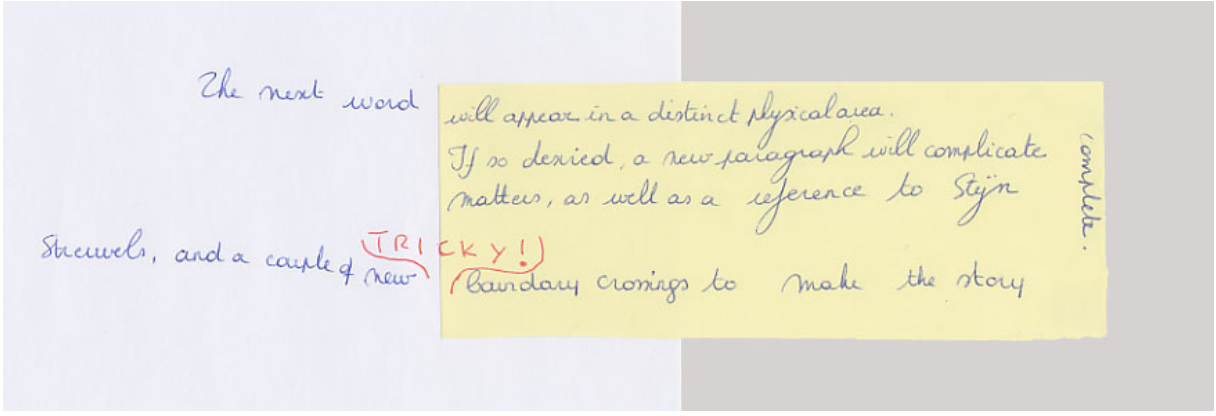
```
<a>...
  <b>...</b>
</a>
```

Overlapping start and end tags, like in the following example, are illegal:

```
<a>...
  <b>...
</a>
  </b>
```

In contrast to XML documents, real-life texts seldom satisfy this qualification. Still, with a minimal degree of abstraction, many textual features can be mapped onto properly nesting XML structures without great difficulties. Yet, (modern) primary manuscript materials have a very *concrete* nature, containing all of the peculiarities of the draft stage like deletions, scribbles, scars,... that are filtered out from the final form of nicely laid-out print texts. Since the TEI scheme is aimed preliminary at the representation of logical structure, problems of overlap will easily appear when encoding the logical and physical aspects of manuscript materials.

The following example illustrates some of the complexities that can arise out of the intricate blends of logical and physical structures:



This example contains logical-physical structures (2 paragraphs, 1 addition), mixed with more logical structures (1 name), and purely physical structures (the shifts

in writing direction and paper). Because of the various overlaps, the encoding of this fragment according to the TEI scheme is not straightforward. Chapter [31. Multiple Hierarchies] of the TEI P4 Guidelines suggests some strategies to overcome the problem of overlap in the XML version of the TEI DTD. They can be summarised along 3 principles:

- One is the use of empty elements 'to mark the beginnings and endings of regions of the text which have something in common'. As these elements do not contain the text region as their content, they can interfere with other hierarchical units without causing overlap. Although logically transparent, this approach encumbers the processing of these structures by XML parsers and other software.
- Another possibility is to use start and end tags for all these different structural elements, but to avoid overlap by splitting up overlapping structures into several elements. This complicates the encoding and doesn't allow straightforward processing either.
- A last option consists of multiple encodings of the same information. Clearly this avoids the problem of competing hierarchies, but can double the encoding effort for every structural view that is to be discerned, and does not allow very practical processing either.

[5]

Applied to this (admittedly, rather complex, but still short) example, it is clear that options 2 and 3 would proof very unwieldy, resulting either in a plethora of split structures or different encodings. Although syntactically correct from a technical point of view, neither of these approaches seems to produce a faithful representation in the markup of a theoretical view on text that sees those structural levels functioning in a continuum. Therefore, in the design of the DALF DTD, we opted for the first solution: encoding overlapping hierarchies with empty elements. In view of the difficulties that exist with the processing of empty elements, this strategy has to be minimised as much as possible, however. The above-mentioned bias of the TEI scheme towards logical structures provided a good rationale. Most of the logical structures in the example can be encoded with existing TEI elements. No TEI elements exist however to mark the start and end of physical materials that contain those logical structures. The **<pb />** TEI element (see [the TEI P4 Guidelines]) is not adequate as its semantics restrict it to the marking of page sides in a book. Therefore, the DALF DTD has been augmented with a mechanism to express a minimal 'layer' view on documents, in a way that avoids both the possibility of overlap with the encoding of

.....

[5] None of these options provides 100% satisfactory solutions, however. Encoding multiple hierarchies in XML has become a generally acknowledged problem for which a number of other solutions (independent of TEI) are proposed, varying from degrees of 'virtual markup' (e.g. standoff-markup, Bottom-Up Virtual Hierarchies) to abandoning XML for another encoding scheme (MECS, TexMECS, Just-In-Time Trees). Up to now, this has not lead to ready-to-use, generally accepted ways of dealing with overlapping hierarchies in XML. Therefore, we stuck to the proposals within the TEI paradigm, which provides after all the best guarantee for future standard-compliance.

logical structures, and also an overgeneralised use of empty elements. This means that a letter can be seen as a complex of physical layers, pieces of physical containers for the logical structures. When different layers are distinguished, their boundaries can be indicated with empty elements that do not disturb the proper nesting of other elements.

This 'layer' approach is implemented in the DALF DTD as a mechanism for identifying and referring to different physical layers, modelled after the TEI mechanism for identifying and referring to document hands (see [the TEI P4 Guidelines]). In order to identify physical layers, the **<layerList>** element is added to the **<profileDesc>** element in the header. It lists different layer definitions as distinct **<layer>** elements:

<layerList>
Contains a definition of the different physical layers in the source document.

<layer>
Identifies each different physical layer the encoder wants to discern. If desired, the encoder can provide a description inside zero or more **<p>** elements (see [the TEI P4 Guidelines]).

The **<layerList>** element has only the global attributes (see 3.3 Global attributes (p. 8)). Each **<layer>** element can have one additional attribute:

type
Characterises the element in some sense, using any convenient classification scheme or typology. It is an optional attribute, with as value type CDATA.

In order to provide some categorisation of the type of layer as an attribute value, DALF encoders are strongly encouraged to provide each layer definition with a **type** attribute, with appropriate values such as 'post-it', 'newspaper_article', etc. Furthermore, although the **id** attribute is never obligatory, it *must* be present on each **<layer>** element, in order to provide a reference point for the elements in the text that signal the boundaries of different physical layers in the document.

For the above-mentioned example, the physical layers and unique hands can be defined in the following way:

```
<profileDesc>
  <handList>
    <hand id="hand2" />
  </handList>
  <layerList>
    <layer id="l2" type="post-it" />
  </layerList>
</profileDesc>
```

When different layers are defined in the header of a DALF letter, some special text

elements can be used to refer to those definitions. Because of the above-mentioned difficulties with overlapping structures and the processing of empty elements, we opted against a ‘milestone’ use of empty elements. Instead, a pair of tags is specified to mark the start and end of physical boundaries, that semantically function as start and end tags, but syntactically are empty elements. They are respectively **<layerStart />** and **<layerEnd />**, and since shifts of physical layers can occur at any place in a letter, they are specified as members of the TEI element class **Incl** (see 7. Modifications to TEI element classes (p. 69)).

<layerStart />
Marks the start of a different physical layer in a letter.

<layerEnd />
Marks the end of a different physical layer in a letter.

In order to refer to a definition of the layer in the header, a special attribute is added to the global ones (defined at 3.3 Global attributes (p. 8)):

layer
Provides a pointer to the definition of the physical layer in the header. This is a required attribute, with IDREF as value type. Its value must thus be the name of an identifier> of a **<layer>** element the header (see 6. Correlations of logical and physical structures (p. 63)).

The TEI scheme provides several means to link fragmented information (see [the TEI P4 Guidelines]). Although these do not provide any straightforward functionality as yet, some pieces of specialised software could make use of them to reconstruct alternative views of the document. In the spirit of the suggestions made in chapter [31. Multiple Hierarchies] of the TEI P4 Guidelines, the following example shows how the physical layers in the example at the start of this section can be encoded, and how a link between those layer elements can be made explicit with the TEI **<join>** element:

```
<text>
  <body>
    <p>The next word <layerStart layer="l2" id="ls1" />will appear
    in a distinct physical area.</p>
    <p>If so desired, a new paragraph will complicate matters, as
    well as a reference to <name>Stijn <layerEnd layer="l2" id="le1" />
    Streuvels</name>, and a couple of new <add hand="hand2">tri
    <layerStart layer="l2" id="ls1b" />cky!</add> boundary crossings
    to make the story <seg rend="90">com<layerEnd layer="l2" id="le1b" />plete.
    </seg></p>
    ...
  </body>
  <back>
    <join targets="ls1 le1 ls1b le1b" result="div" desc="physical layer" />
  </back>
</text>
```

The following lines demonstrate how the DALFExtns.dtd file redefines the **<profileDesc>** element to include **<layerList>**, and how the unique DALF elements used to identify and refer to physical layers in a letter are defined:

```
<!ELEMENT profileDesc %om.RR;
        (creation?, langUsage*, textDesc*,particDesc*, settingDesc*,
        handList*, layerList*, textClass*)>
<!ATTLIST profileDesc
        %a.global;>

<!ELEMENT layerList %om.RR;
        (layer*)>
<!ATTLIST layerList
        %a.global;>

<!ELEMENT layer %om.RR;
        (p)*>
<!ATTLIST layer
        %a.global;
        type CDATA #IMPLIED>

<!ELEMENT layerStart %om.RR;
        EMPTY>
<!ATTLIST layerStart
        %a.global;
        layer IDREF #REQUIRED>

<!ELEMENT layerEnd %om.RR;
        EMPTY>
<!ATTLIST layerEnd
        %a.global;
        layer IDREF #REQUIRED>
```

7. Modifications to TEI element classes

In the course of test-tagging real correspondence material with a draft of the DALF DTD, the distribution of some TEI elements proved too narrow to cater for the encoding of the corresponding textual phenomena where they occur. On a documentary level, additions for example can occur before any other textual element in a letter, and closing formulae can contain notes or note markers. However, TEI elements like **<add>** and **<note>** are not allowed in those contexts.

Since we aim at a faithful representation of the documentary source with the DALF encoding scheme, we followed the suggestions made on various occasions in the TEI public mailing list [6] (see [the TEI P4 Guidelines]), namely to redefine some TEI elements as global elements. Technically, this solution is explained in great detail in Syd Bauman's message posted on 10 Juni 2002, and illustrated in the documentation of the changes regarding element class manipulations in the Women Writers Project (see [the TEI P4 Guidelines]).

The DALF encoding scheme redefines the following TEI elements as global elements: A first step in this process consists of the extension of the x-dot entity for the **Incl** element class with these elements, as explained in section [3.7.2. Classes Used in Content Models] and section [29.1.3. Class Extension] of the TEI P4 Guidelines. Together with the other global DALF elements already discussed in these guidelines, the new DALF declaration for the **Incl** element class in the file DALFExtns.ent looks as follows:

```
<!ENTITY % x.Incl 'add | deco | figure | paraph | print | layerStart | layerEnd | note | seg |'>
<!ENTITY % m.Incl "%x.Incl; %n.anchor; | %m.editIncl; | %m.metadata; | %m.refsys;">
```

Contrary to the declaration of new DALF elements as members of **Incl**, the redefinition of already existing TEI elements can imply several other changes, such as the removal of those elements from the classes and elements where they are defined in the TEI DTD, in order to avoid the danger of an invalid DTD with lots of "ambiguous content models". The next sections discuss the motivations for the redefinition of the TEI elements mentioned, and the changes introduced by this process.

7.1 Changes relating to the declaration of <add> as global element

.....

[6] In particular, the "suggestions for the forthcoming/ongoing (?) TEI revision" made by Hilde Bøe, Ellen Nessheim and Stine Brenna Taugbø for the Henrik Ibsen's Writings, posted on 4 December 2001. These included (a.o.) the redefenition of <**add**>, <**figure**>, and <**note**> as global elements. Also in his answer of 28 July 2000 to a question regarding the impossibility to use <**note**> inside <**salute**>, Syd Bauman describes this as "an error with TEI including the TEI-Lite view", and suggests the option to extend the TEI DTD.

Additions can occur anywhere on a letter, since additions originate from a "second-level" writing act, in which the "secondary" author does not necessarily consider the original letter as such. Additions are not necessarily written in "letter mode" and thus not necessarily follow its structural conventions.

In the TEI DTD, **<add>** is defined as member of the **edit** element class. Consequently, the declaration of **<add>** as member of the more encompassing **Incl** element class in the DALF DTD mandates its deletion from **edit**. Therefore, the **edit** element class is redefined in the file DALFExtns.ent in the following way:

```
<!ENTITY % m.edit "app | corr | damage | del | orig | reg | restore | sic | space | supplied |
unclear">
```

7.2 Changes relating to the declaration of <figure> as global element

Figures can occur in "bare" form (i.e. outside other containing **<body>**-level elements) anywhere on a letter, since in some cases they originate from a "non-letter mode", e.g. when a distracted author/receiver makes marginal drawings that are (rather) unrelated to the contents and/or logical structure of the letter.

In the TEI DTD, **<figure>** is defined as member of the **common**[7], **inter**, and **tpParts** element classes. The declaration of **<figure>** as member of the **Incl** element class in the DALF DTD means that the **common**, **inter** and **tpParts** element classes should be redefined without it. The following lines illustrate these changes made in the file DALFExtns.ent:

```
<!ENTITY % x.common 'calc |'>
<!ENTITY % m.common " calc | bibl | biblFull | biblStruct | ab | eTree | graph | l | lg | p | sp |
        tree | witList | cit | q | quote | label | list | listBibl  | witDetail | stage | table">

<!ENTITY % x.inter 'calc |'>
<!ENTITY % m.inter "calc | bibl | biblFull | biblStruct | castList | cit | q | quote |
        label | list | listBibl | witDetail | stage | camera | caption | move | sound |
        tech | view | table | text">

<!ENTITY % x.tpParts 'calc |'>
<!ENTITY % m.tpParts "%x.tpParts; %n.byline; | %n.docAuthor; | %n.docDate; |
        %n.docEdition; | %n.docImprint; | %n.docTitle; | %n.epigraph; |
        %n.figure; | %n.imprimatur; | %n.titlePart;">
```

7.3 Changes relating to the declaration of <note> as global element

.....

[7] See note 3 for a discussion on the distribution of the TEI <**figure**> element.

Authorial notes will only occur inside other text elements, yet in broader contexts than those defined by the TEI element class **notes**. For example, an author may make a note to the address line of an opening formula, or to the title of a work mentioned in the letter.

In the TEI DTD, **<note>** is defined as member of the **biblPart**, **notes**, **terminologyInclusions**, and **dictionaryTopLevel** element classes. Of these, only the first two are used in the DALF DTD. The transformation of **<note>** to a global element thus involved its removal from the classes **biblPart** and **notes**. Those element classes are defined in the file DALFExtns.ent as follows:

```
<!ENTITY % m.biblPart "analytic | author | biblScope | edition | editor | extent |
        idno | imprint | monogr | pubPlace | publisher | respStmt | series">

<!ENTITY % m.notes "witDetail">
```

A global **<note>** element caused further difficulties with the content models of some other TEI elements. The original TEI declarations of the elements **<biblStruct>** and **<monogr>** explicitly mentioned **<note>** in their content models. With the declaration of **<note>** as member of the **Incl** element class those content models became ambiguous. Therefore, they were redefined in the file DALFExtns.dtd without the **<note>** element in the following way:

```
<!ELEMENT biblStruct %om.RO;
        ((%m.Incl;)*, (analytic, (%m.Incl;)*)?,
        ((monogr, (%m.Incl;)*), (series, (%m.Incl;)*)*)+,
        (idno, (%m.Incl;)*)*)>
<!ATTLIST biblStruct
        %a.global;
        %a.declarable;>

<!ELEMENT monogr %om.RO;
        (( (%m.Incl;)*,
            ((
              (author | editor | respStmt),
              (author | editor | respStmt | %m.Incl;)*,
              (title, (%m.Incl;)*)+,
              ( (editor | respStmt), (%m.Incl;)* )* )
            |(
              (title, (%m.Incl;)*)+,
              ((author | editor | respStmt), (%m.Incl;)*)*
            )))?,
            ((meeting), (%m.Incl;)*)*,
            (edition, (editor | respStmt | %m.Incl;)*, imprint,
            (imprint | extent | biblScope | %m.Incl;)*
           )>
<!ATTLIST monogr
        %a.global;>
```

7.4 Changes relating to the declaration of <seg> as global element

An encoding scheme for primary manuscript materials should be able to cater for the documentary transcription of phenomena as they occur. Of course, there may be as many different structures as there are letters one may wish to encode. Yet, an encoding scheme that would anticipate this by allowing every element in the broadest possible contexts, would be minimally expressive. A scheme with too many global elements would not add much more to structural insight than the letter itself. Therefore, some level of abstraction should be maintained. This rationale motivated the declaration of **<seg>** as global element in the DALF DTD. Its multi-purpose semantics, namely "to mark any segments of the text of interest for processing", and its general content model make it an interesting intermediary global container element that can wrap up other loose text or text elements on places where no text or other elements are allowed. In this way, the number of global elements is not over-exaggerated, while the expressive capacity of the encoding scheme is adapted to the irregular structure of the documents it is designed to represent.

In the TEI DTD, **<seg>** is defined as member of the **seg** element class. Consequently, the declaration of **<seg>** as member of the more encompassing **Incl** element class in the DALF DTD mandates its deletion from **seg**. The following lines illustrate this change made in the file DALFExtns.ent:

```
<!ENTITY % m.seg "c | cl | m | phr | s | w">
```

A global **<seg>** element caused further difficulties with the content models of another TEI element. The original TEI declaration of the element **<sp>** explicitly included **<seg>** in its content model. With the declaration of **<seg>** as member of the **Incl** element class that content model became ambiguous. Therefore, it was redefined in the file DALFExtns.dtd without the **<seg>** element in the following way:

```
<!ELEMENT sp %om.RO;
        ((%m.Incl;)*, (speaker, (%m.Incl;)*)?,((p | l | lg | ab | stage), (%m.Incl;)*)+)>
<!ATTLIST sp
        %a.global;
        who IDREFS #IMPLIED>
```

Appendix A. Reference documentation for DALF elements and classes

Appendix A.1 Elements

Appendix A.1.a DALF elements

A.1.a.1 <accMat>

Description Contains details of any significant additional material which may be closely associated with the letter.

Attributes In addition to global (p. 141) attributes:

type Describes the type of additional material.

Datatype: CDATA

Values: Any convenient typology can be used. Sensible values may be ‘external_document’, ‘tax_form’, etc.

Default: #IMPLIED

Example

```
<accMat type="external_document">
  <p>A copy of a tax form from 1947 is included in the envelope
  with the letter. It is not catalogued separately.</p>
</accMat>
```

Content Must contain one or more paragraphs.

Parents additional

Children p

Declaration

```
<!ELEMENT accMat %om.RR;
        (p+)>
<!ATTLIST accMat
        %a.global;
        type CDATA #IMPLIED>
```

See further 4.7.3 Accompanying materials: **<accMat>** (p. 42)

A.1.a.2 <acquisition>

Description Contains any descriptive or other information concerning the process by which a letter entered the holding institution.

Attributes No other than global (p. 141) attributes

Example

```
<acquisition>
  <p>acquired by the AMVC through a gift by the heirs of the whole archive</p>
</acquisition>
```

Note This element is reserved solely for the description of the acquisition process by which a letter entered the holding institution. It does not cover facts about the origination of a letter (see **<origin>** (p. 110), its history before it entered the holding institution (see **<provenance>** (p. ?)), or its history after it entered the holding institution (see **<custodialHist>** (p. 83)).

The historical events should be listed in chronological order.

Content Must contain one or more paragraphs.

Parents history

Children p

Declaration

```
<!ELEMENT acquisition %om.RR;
        (p+)>
<!ATTLIST acquisition
        %a.global;>
```

See further 4.6 The history of the letter: **<history>** (p. 35)

A.1.a.3 <additional>

Description Groups additional information about the letter.

Attributes No other than global (p. 141) attributes

Example

```
<additional>
  <adminInfo>
    <availability>
      <p>Available under licence from the publishers.</p>
    </availability>
```

```
  </adminInfo>
  <surrogates>
    <p>a microfilm facsimile exists</p>
  </surrogates>
  <accMat>
    <p>half a page of the "De Gentenaar" journal of 16 January
    1932 has been included in the envelope</p>
  </accMat>
  <listBibl>
    <bibl>This letter is included in
      <author>Marcel De Smedt &amp;amp; Edward Vanhoutte</author>
      <title>Stijn Streuvels, De Teleurgang van den Waterhoek.
      Elektronisch-kritische editie/electronic-critical edition.</title>
      <pubPlace>Amsterdam</pubPlace>
      <publisher>Amsterdam University Press/KANTL</publisher>
      <date>2000</date>
      <idno type="ISBN">90-5356-441-1 (CD-Rom)</idno>
    </bibl>
  </listBibl>
</additional>
```

Content Must contain at least one of the child elements; when more are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children accMat adminInfo listBibl note surrogates

Declaration

```
<!ELEMENT additional %om.RR;
        ((adminInfo, surrogates?, accMat?, listBibl?, note*) |
        (surrogates, accMat?, listBibl?, note*) |
        (accMat, listBibl?, note*) | (listBibl, note*))>
<!ATTLIST additional
        %a.global;>
```

See further 4.7 Additional information: **<additional>** (p. 37)

A.1.a.4 <addressee>

Description Identifies a/the person to whom the letter was addressed.

Attributes In addition to global (p. 141) attributes and those inherited from names (p. ?) :

attested Indicates the status of the contents of the element on which it appears with regard to the evidence from which it is derived.

Datatype: (yes | no | added | unk)

Legal values are:

‘yes’ the attribution in the element is made on the basis of evidence inside the letter

‘added’ the attribution in the element is made on the basis of additional material accompanying the letter (eg. the envelope)

‘no’ the attribution in the element is made on the basis of external evidence

‘unk’ it is unknown on what basis the attribution in the element is made

Default: #IMPLIED

accepted Indicates whether the attribution made in the element is generally accepted. This is an optional attribute with one of following three values. Generally accepted attributions are encoded with value "yes"; attributions that are not generally accepted with "no"; when no claim can be made regarding this status, the value "unk" is used.

Datatype: (yes | no | unk)

Legal values are:

‘yes’ the attribution in the element is generally accepted

‘no’ the attribution in the element is not generally accepted

‘unk’ it is unknown whether the attribution in the element is generally accepted

Default: #IMPLIED

Example

```
<addressee attested="add">Maurice De Meyer</addressee>
```

Class names; typed

Content May contain character data and phrase-level elements.

Parents letHeading

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp

interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT addressee %om.RR;
        (%phrase.seq;)>
<!ATTLIST addressee
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED
        accepted (yes | no | unk) #IMPLIED>
```

See further 4.2.1 The communicative participants: **<author>**, **<addressee>** and **<respStmt>** (p. 20)

A.1.a.5 <adminInfo>

Description Contains information about the present custody and availability of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<adminInfo>
  <availability status="restricted">
    <p>Available for academic research purposes only.</p>
  </availability>
  <custodialHist>
    <custEvent type="loan">
      <p>from 13 January 1955 to 3 March 1956, the letter was given on loan to
      the university of Amsterdam</p>
    </custEvent>
  </custodialHist>
  <remarks>
    <p>maybe this letter can turn out significant for the retracing of the
    missing letters by Streuvels to Excelsior between 1926 and 1928</p>
  </remarks>
</adminInfo>
```

Content Must contain at least one of the child elements; when more are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents additional

Children availability custodialHist note remarks

Declaration

```
<!ELEMENT adminInfo %om.RR;
        ((availability, custodialHist?, remarks?, note*) |
        (custodialHist, remarks?, note*) | (remarks, note*))>
<!ATTLIST adminInfo
        %a.global;>
```

See further 4.7.1 Administrative information: **<adminInfo>** (p. 39)

A.1.a.6 <altName>

Description Contains any form of alternative identifier used for a document. This may be a nickname or a former identifier.

Attributes Global (p. 141) attributes and those inherited from names (p. 145) and typed (p. 148)

Example

```
<altName type="former_id">SL410811/b</altName>
<altName type="nickname">the birthday letter</altName>
```

Class names; typed

Content May contain character data and phrase-level elements.

Parents letIdentifier

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT altName %om.RR;
        (%phrase.seq;)>
<!ATTLIST altName
        %a.global;
        %a.names;
        %a.typed;>
```

See further 4.1.2 The micro-location path of a letter: **<collection>**, **<idno>** and **<altName>** (p. 17)

A.1.a.7 <arg>

Description Contains the argument of a calculation.

Attributes No other than global (p. 141) attributes

Example

```
<arg>
  <arg>5</arg>
  <add>
    <oper>+</oper>
    <arg>(<arg>67</arg> <oper>/</oper> <arg>4</arg>)</arg>
  </add>
</arg>
```

Note An argument can be defined as the left-hand side of an equation in a calculation. It can thus only occur as a descendant of a **<calc>** (p. 79) element. An argument may be in strictly numerical form, or may be intermingled with textual data.

Content May contain character data, phrase-level elements and embedded calculations and arguments.

Parents arg calc result

Children #PCDATA abbr add addSpan address alt altGrp anchor app arg calc cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT arg %om.RR;
        (%phrase; | calc | arg)*>
<!ATTLIST arg
        %a.global;>
```

See further 5.3 Calculations: **<calc>** (p. 52)

A.1.a.8 <calc>

Description Contains a calculation.

Attributes No other than global (p. 141) attributes

Example

```
<calc>
  <arg>969 <abbr expan="exemplaren">ex.</abbr> (zie afrekening van
```

```
30.8.41)</arg>
  <oper>-</oper>
  <arg>138<abbr expan="exemplaren">ex</abbr> (
    <calc>
      <arg>133 <abbr expan="exemplaren">ex.</abbr> verkocht</arg>
      <oper>+</oper>
      <arg>5<abbr expan="persexemplaren">persex.</abbr></arg>
    </calc>)
  </arg>
  <result><hi rend="double_underlined">831</hi><abbr
expan="exemplaren">ex.</abbr></result>
</calc>
```

Note A calculation may be in strictly numerical form, or may be intermingled with textual data.

Class common; inter; tpParts

Content May contain character data, phrase-level elements, embedded calculations, and arguments, operators and results.

Parents ab add arg argument body calc camera caption castList cell corr country damage div div0 div1 div2 div3 div4 div5 div6 div7 docEdition emph epigraph epilogue figDesc foreign head hi imprimatur item l lem meeting metDecl note p performance print prologue ps q quote rdg ref region rendition result seg set sic sound stage supplied tagUsage tech title titlePage titlePart unclear view wit witDetail witness xref

Children #PCDATA abbr add addSpan address alt altGrp anchor app arg calc cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num oper orig paraph pb placeName print ptr ref reg restore result rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT calc %om.RR;
        (%phrase; | calc | arg | oper | result)*>
<!ATTLIST calc
        %a.global;>
```

See further 5.3 Calculations: **<calc>** (p. 52)

A.1.a.9 <class>

Description Contains a functional-communicative "genre"-indication of letters (still to

be developed).

Attributes In addition to global (p. 141) attributes:

type Characterises the class in some sense.

Datatype: CDATA

Values: Any convenient classification scheme or typology may be used.

Default: #IMPLIED

Example

```
<class>love letter</class>
```

Content May contain character data only.

Parents letContents

Children #PCDATA

Declaration

```
<!ELEMENT class %om.RR;
        (#PCDATA)>
<!ATTLIST class
        %a.global;
        type CDATA #IMPLIED>
```

See further 4.5 The contents of the letter: **<letContents>** (p. 34)

A.1.a.10 <collection>

Description Contains an identification of a collection of documents of which the letter forms part, not necessarily located within a single repository.

Attributes No other than global (p. 141) attributes and those inherited from names (p. 145) and typed (p. 148).

Example

```
<collection key="SL/AMVC">Streuvels-Lannoo</collection>
```

Class names; typed

Content May contain character data and phrase-level elements.

Parents letIdentifier

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT collection %om.RR;
        (%phrase.seq;)>
<!ATTLIST collection
        %a.global;
        %a.names;
        %a.typed;>
```

See further 4.1.2 The micro-location path of a letter: **<collection>**, **<idno>** and **<altName>** (p. 17)

A.1.a.11 <condition>

Description Describes the physical condition of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<condition>
  <p>topright corner is missing</p>
  <p>the folding lines have caused some tears</p>
</condition>
```

Content Must contain one or more paragraphs.

Parents physDesc

Children p

Declaration

```
<!ELEMENT condition %om.RR;
        (p+) >
<!ATTLIST condition
        %a.global;>
```

See further 4.3.7 The physical condition: **<condition>** (p. 32)

A.1.a.12 <custEvent>

Description Describes a single event related to the custodial history of a letter.

Attributes In addition to global (p. 141) attributes :

type Characterises the type of historical event.

Datatype: CDATA

Values: Any convenient typology can be used. Sensible values may be ‘restoration’, ‘loan’, etc.

Default: #IMPLIED

Example

```
<custEvent type="loan">
  <p>from 13 January 1955 to 3 March 1956, the letter was given on loan to
  the university of Amsterdam</p>
</custEvent>
<custEvent type="restoration">
  <p>during the summer of 1978, the letter has been restored</p>
</custEvent>
```

Note The historical events should be listed in chronological order.

Content Must contain one or more paragraphs.

Parents custodialHist

Children p

Declaration

```
<!ELEMENT custEvent %om.RR;
        (p+)>
<!ATTLIST custEvent
        %a.global;
        type CDATA #IMPLIED>
```

See further 4.7.1 Administrative information: **<adminInfo>** (p. 39)

A.1.a.13 <custodialHist>

Description Contains a description of the custodial history of a letter

Attributes No other than global (p. 141) attributes

Example

```
<custodialHist>
  <custEvent type="loan">
    <p>from 13 January 1955 to 3 March 1956, the letter was given on loan to
```

```
    the university of Amsterdam</p>
  </custEvent>
  <custEvent type="restoration">
    <p>during the summer of 1978, the letter has been restored</p>
  </custEvent>
</custodialHist>
```

Note This element is reserved solely for the description of the history of a letter after it entered the holding institution. It does not cover facts about the origination of a letter (see origin (p. 110), its history before it entered the holding institution (see provenance (p. 121)), or the acquisition process by which a letter entered the holding institution (see acquisition (p. 74)).

Content Must contain at least one child element. The note(s) may only be used to provide additional information after all other elements.

Parents adminInfo

Children custEvent note

Declaration

```
<!ELEMENT custodialHist %om.RR;
        (custEvent+, note*)>
<!ATTLIST custodialHist
        %a.global;>
```

See further 4.7.1 Administrative information: **<adminInfo>** (p. 39)

A.1.a.14 <dateLet>

Description Identifies the date when the letter was written.

Attributes In addition to global (p. 141) attributes:

attested Indicates the status of the contents of the element on which it appears with regard to the evidence from which it is derived.

Datatype: (yes | no | added | unk)

Legal values are:

'yes' the attribution in the element is made on the basis of evidence inside the letter

'added' the attribution in the element is made on the basis of additional material accompanying the letter (eg. the envelope)

'no' the attribution in the element is made on the basis

of external evidence

'unk' it is unknown on what basis the attribution in the element is made

Default: #IMPLIED

Example

```
<dateLet attested="no">1945-01-13</dateLet>
```

Content May contain character data and phrase-level elements.

Parents letHeading

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT dateLet %om.RR;
        (%phrase.seq;)>
<!ATTLIST dateLet
        %a.global;
        attested (yes | added | no | unk) #IMPLIED>
```

See further 4.2.2 The place and time of writing: **<placeLet>** and **<dateLet>** (p. 23)

A.1.a.15 <deco />

Description Marks the occurrence of a decoration, ie. a decorative element that has originated within the writing act.

Attributes In addition to global (p. 141) attributes:

decoRef refers to a previously defined decoration

Datatype: IDREF

Values: must be one of the decoration identifiers declared in the header (see A.1.a.20 **<decoration>** (p. 89))

Default: #REQUIRED

entity names the external entity within which the graphic image of the figure is stored.

Datatype: ENTITY

Default: #IMPLIED

Example

```
<?xml version="1.0"?>
<!DOCTYPE TEI.2 PUBLIC "//CTB//DTD Dalf 1.0 (based on TEI)//NL"
"DALF.dtd" [
<!NOTATION jpeg  PUBLIC
   'ISO DIS 10918//NOTATION JPEG Graphics Format//EN' >
<!NOTATION png PUBLIC
   '-//TEI//NOTATION IETF RFC2083 Portable Network Graphics//EN'>
<!ENTITY fig1 SYSTEM "fig1.jpeg" NDATA jpeg>
<!ENTITY fig2 SYSTEM "fig2.jpeg" NDATA jpeg>
]>
<TEI.2>
  <text>
    ...
    <p>Een grote boom stond daar <deco decoRef="fig1" entity="fig1" />...</p>
    ...
    <p>Ik stel mij de pagina zo voor: <deco decoRef="fig2" entity="fig2" /></p>
    ...
  </text>
</TEI.2>
```

Class Incl

Content This is an empty element.

Parents ab abbr actor add addrLine address addressee altName analytic app arg argument author authority back bibl biblFull biblScope biblStruct bloc body byline calc camera caption castGroup castItem castList cell cit classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor div div0 div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle edition editor emph envPart epigraph epilogue expan extent figDesc figure foreign front funder fw gloss group head headItem headLabel height hi imprimatur imprint institution item l label language lem lg list listBibl measure meeting mentioned metDecl monogr name note num opener oper orig p performance placeLet placeName principal print prologue ps pubPlace publicationStmt publisher q quote rdg rdgGrp ref reg region rendition repository resp respStmt restore result role roleDesc row rs salute seg series set settlement sic signed soCalled sound sp speaker sponsor stage street supplied symbol table tagUsage tech term text time timeRange title

titlePage titlePart trailer unclear view width wit witDetail witList witness xref

Children EMPTY

Declaration

```
<!ELEMENT deco %om.RR;
        EMPTY>
<!ATTLIST deco
        %a.global;
        decoRef IDREF #REQUIRED
        entity ENTITY #IMPLIED>
```

See further 5.5 Decorative elements: **<deco />** and **<paraph />** (p. 57)

A.1.a.16 <decoDesc>

Description Contains a description of a decoration in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<decoDesc>
  <p>small black/white drawing of a tree</p>
</decoDesc>
```

Content Must contain one or more paragraphs.

Parents decoItem

Children p

Declaration

```
<!ELEMENT decoDesc %om.RR;
        (p+)>
<!ATTLIST decoDesc
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.17 <decoItem>

Description Contains a structured description of a decoration in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<decoItem id="fig2">
```

```
  <decoDesc>
    <p>small color drawing of suggested page layout, containing some text</p>
  </decoDesc>
  <decoText>
    <p>Eerste paragraaf Hier</p>
    <p>voetnoten!</p>
  </decoText>
</decoItem>
```

Note The **<decoItem>** element is used in the header to define each decoration that appears in the letter. One such element must appear within the header for each decoration distinguished in the text.

Although the global id attribute is never obligatory, it *must* be present on each **<decoItem>** element, in order to provide a reference point for the **<deco />** (p. 85) element in the text that signals the occurrence of the decoration.

Content Must contain at least a description of the decoration, possibly followed by a transcription of text contained in the decoration.

Parents decoList

Children decoDesc decoText

Declaration

```
<!ELEMENT decoItem %om.RR;
        (decoDesc, decoText?)>
<!ATTLIST decoItem
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.18 <decoList>

Description Contains a list of structured descriptions for each of the decorations in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<decoList>
  <decoItem id="fig1">
    <decoDesc>
      <p>small black/white drawing of a tree</p>
    </decoDesc>
  </decoItem>
  <decoItem id="fig2">
    <decoDesc>
```

```
      <p>small color drawing of suggested page layout, containing some text</p>
    </decoDesc>
    <decoText>
      <p>Eerste paragraaf Hier</p>
      <p>voetnoten!</p>
    </decoText>
  </decoItem>
</decoList>
```

Content Must contain at least one list item.

Parents decoration

Children decoItem

Declaration

```
<!ELEMENT decoList %om.RR;
        (decoItem+)>
<!ATTLIST decoList
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.19 <decoText>

Description Contains a rendering of text contained in a decoration.

Attributes No other than global (p. 141) attributes

Example

```
<decoText>
  <p>Eerste paragraaf Hier</p>
  <p>voetnoten!</p>
</decoText>
```

Content Must contain one or more paragraphs.

Parents decoItem

Children p

Declaration

```
<!ELEMENT decoText %om.RR;
        (p+)>
<!ATTLIST decoText
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.20 <decoration>

Description Describes decorative items in the letter that have originated within the writing act.

Attributes No other than global (p. 141) attributes

Example

```
<decoration>
  <decoList>
    <decoItem id="fig1">
      <decoDesc>
        <p>small black/white drawing of a tree</p>
      </decoDesc>
    </decoItem>
    <decoItem id="fig2">
      <decoDesc>
        <p>small color drawing of suggested page layout, containing some text</p>
      </decoDesc>
      <decoText>
        <p>Eerste paragraaf Hier</p>
        <p>voetnoten!</p>
      </decoText>
    </decoItem>
  </decoList>
</decoration>
```

Example

```
<decoration>
  <p>small black/white drawing of a tree</p>
  <p>small color drawing of suggested page layout, containing the text
    <q>Eerste paragraaf Hier <lb /> voetnoten!</q>
  </p>
</decoration>
```

Note The decorations in a document can be described either in a loose prose description, or in a list (recommended), containing a structured definition per decorative item concerned.

Content A prose description of the decorations should contain one or more paragraphs; a structured description must be done in a single **<decoList>** element.

Parents physDesc

Children decoList p

Declaration

```
<!ELEMENT decoration %om.RR;
        (decoList | p+)>
<!ATTLIST decoration
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.21 <depth>

Description Contains a measurement for the depth of an object.

Attributes No other than global (p. 141) attributes and those inherited from measured (p. 139)

Example

```
... het bad is <dimensions>
   <depth units="m">1,24 m</depth> diep
</dimensions>
```

Class measured

Content May contain character data and phrase-level elements.

Parents dimensions

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT depth %om.RR;
        (%phrase;)*>
<!ATTLIST depth
        %a.global;
        %a.measured;>
```

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

A.1.a.22 <dimensions>

Description Contains any kind of dimensional specification.

Attributes In addition to global (p. 141) attributes and those inherited from measured (p. 139):

type Describes the type of dimensional specification.

Datatype: CDATA

Values: Any convenient typology can be used.

Default: #IMPLIED

Example

```
... het kaft is <dimensions>
   <height units="cm">40 cm</height>
   hoog en
   <width units="cm">25 cm</width>
   breed
</dimensions>
```

Class measured; phrase

Content May contain character data, phrase-level elements, and specific elements for the specification of height, width and depth.

Parents ab abbr actor add addrLine addressee altName arg author authority bibl biblScope bloc byline calc camera caption castItem catDesc cell classCode closer collection corr country creation damage date dateLet dateRange del depth dimensions distance distinct distributor docAuthor docDate docEdition docImprint edition editor emph expan extent figDesc foreign funder fw gloss head headItem headLabel height hi imprimatur institution item l label language lem measure meeting mentioned name note num opener oper orig p placeLet placeName principal print ps pubPlace publisher q quote rdg ref reg region rendition repository resp restore result role roleDesc rs salute seg settlement sic signed soCalled sound speaker sponsor stage street supplied symbol tagUsage tech term time timeRange title titlePart trailer unclear view width wit witDetail witness xref

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan depth dimensions distinct emph expan figure foreign formula fw gap gloss handShift height hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear width xptr xref

Declaration

```
<!ELEMENT dimensions %om.RR;
        (%phrase;|height|width|depth)*>
<!ATTLIST dimensions
        %a.global;
        %a.measured;
```

```
        type CDATA #IMPLIED>
```

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

A.1.a.23 <envOcc>

Description Contains an indication of the presence or absence of an envelope.

Attributes In addition to global (p. 141) attributes:

occ Indicates the occurrence of an envelope.

Datatype: (yes | no)

Legal values are:
'yes' the letter is accompanied by an envelope
'no' the letter is not accompanied by an envelope

Default: #REQUIRED

Example

```
<envOcc occ="yes" />
```

Note When an envelope is present, the '<envOcc occ="yes" />' in the header should be matched with the text element **<envelope>** (p. 95) .

The envelope can be functionally defined as 'the part/room/space *reserved for* postal information'. This definition includes the postal information, as well as all other possible contents of that functional part of the letter.

Content This is an empty element.

Parents letDesc

Children EMPTY

Declaration

```
<!ELEMENT envOcc %om.RR;
        EMPTY>
<!ATTLIST envOcc
        %a.global;
        occ (yes | no) #REQUIRED>
```

See further 4.4 Presence of an envelope: **<envOcc />** (p. 33)

A.1.a.24 <envPart>

Description Contains the information on one side of the envelope.

Attributes In addition to global (p. 141) attributes:

side Describes which side of the envelope features the data described.

Datatype: (front | back | postcard)

Legal values are:
'front' the data appears at the front side of the envelope
'back' the data appears at the back side of the envelope
'postcard' the data appears on a one-sided postcard envelope

Default: #IMPLIED

Example

```
<envPart side="front">
  <address type="receiver">
    <addrLine>Stijn Streuvels</addrLine>
    <addrLine>Lijsternest</addrLine>
    <addrLine>Ingooigem</addrLine>
  </address>
  <postmark>
    <placeName>Tielt</placeName>
    <date value="1924-10-24">24.10.'24</date>
  </postmark>
  <div>
    <p>Dank u, postbode!</p>
  </div>
</envPart>
```

Content Must contain at least one of the child elements.

Parents envelope

Children add addSpan address alt altGrp anchor cb deco delSpan div figure fw gap index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp milestone note paraph pb postmark print seg timeline

Declaration

```
<!ELEMENT envPart %om.RR;
        (address | postmark | div | %m.Incl;)+>
<!ATTLIST envPart
        %a.global;
        side (front | back | postcard) #IMPLIED>
```

See further 5.1 The envelope: **<envelope>** (p. 47)

A.1.a.25 <envelope>

Description Contains the information on the envelope.

Attributes In addition to global (p. 141) attributes:

type Describes the type of envelope.

Datatype: CDATA

Values: Any convenient typology can be used. Sensible values may be ‘postcard’, ‘window_envelope’, ‘C5’, ‘160x150’ etc.

Default: #IMPLIED

Example

```
<envelope type="220x110">
  <envPart side="front">
    <address type="receiver">
      <addrLine>Stijn Streuvels</addrLine>
      <addrLine>Lijsternest</addrLine>
      <addrLine>Ingooigem</addrLine>
    </address>
    <postmark>
      <placeName>Tielt</placeName>
      <date value="1924-10-24">24.10.'24</date>
    </postmark>
    <div>
      <p>Dank u, postbode!</p>
    </div>
  </envPart>
  <envPart side="back">
    <address type="sender">
      <addrLine>Lannoo uitgeverij</addrLine>
      <addrLine>Meulebeekschesteenweg 641</addrLine>
      <addrLine>Tielt</addrLine>
    </address>
  </envPart>
  <envelope type="postcard">
    <note>this envelope contains no data</note>
  </envelope>
</envelope>
```

Note The envelope can be functionally defined as ‘the part/room/space *reserved for* postal information’. This definition includes the postal information, as well as all other possible contents of that functional part of the letter.

When an **<envelope>** element is present in the text, this should be matched by a envOcc / (p. 93) element in the header.

Content Must contain at least one of the child elements.

Parents envelope text

Children envPart envelope note

Declaration

```
<!ELEMENT envelope %om.RR;
        (envPart | envelope | note)+>
<!ATTLIST envelope
        %a.global;
        type CDATA #IMPLIED>
```

See further 5.1 The envelope: **<envelope>** (p. 47)

A.1.a.26 <height>

Description Contains a measurement for the height of an object.

Attributes No other than global (p. 141) attributes and those inherited from measured (p. 139)

Example

```
... het kaft is <dimensions>
  <height units="cm">40 cm</height>
  hoog ...
</dimensions>
```

Class measured

Content May contain character data and phrase-level elements.

Parents dimensions

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT height %om.RR;
        (%phrase;)*>
```

```
<!ATTLIST height
        %a.global;
        %a.measured;>
```

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

A.1.a.27 <history>

Description Contains a description of the history of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<history>
  <origin>
    <p>draft of letter found, dated 17 August 1924</p>
  </origin>
  <provenance>
    <p>between January 1925 and 3 September 1935, the letter resided
    in the private collection of the receiver</p>
    <p>between the death of the receiver on 3 September 1935 and 7
    February 1937, the letter was owned by the heirs of the reveiver</p>
  </provenance>
  <acquisition>
    <p>acquired by the AMVC through a gift by the heirs of the whole archive</p>
  </acquisition>
</history>
```

Content Must contain at least one of the child elements; when more are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children acquisition note origin provenance

Declaration

```
<!ELEMENT history %om.RR;
        ((origin, provenance?, acquisition?, note*) | (provenance, acquisition?, note*) |
(acquisition, note*))>
<!ATTLIST history
        %a.global;>
```

See further 4.6 The history of the letter: **<history>** (p. 35)

A.1.a.28 <institution>

Description Contains the name of an organisation within which a manuscript repository is located.

Attributes No other than global (p. 141) attributes and those inherited from names (p. 145) and typed (p. 148)

Example
```
<institution type="municipal_department">Dept. of Culture, Antwerp</institution>
```

Class names; typed

Content May contain character data and phrase-level elements.

Parents letIdentifier

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration
```
<!ELEMENT institution %om.RR;
        (%phrase.seq;)>
<!ATTLIST institution
        %a.global;
        %a.names;
        %a.typed;>
```

See further 4.1.1 The macro-location path of a letter: **<country>**, **<region>**, **<settlement>**, **<institution>** and **<repository>** (p. 15)

A.1.a.29 <layer>

Description Identifies each different physical layer the encoder wants to discern.

Attributes In addition to global (p. 141) attributes:

type Describes the type of physical layer.

Datatype: CDATA

Values: Any convenient typology can be used. Sensible values may be ‘post-it’, ‘sticker’, etc.

Default: #IMPLIED

Example

```
<text>
  <body>
    <p>The next word <layerStart layer="l2" id="ls1" />will appear
    in a distinct physical area.</p>
    <p>If so desired, a new paragraph will complicate matters, as
    well as a reference to <name>Stijn <layerEnd layer="l2" id="le1" />
    Streuvels</name>, and a couple of new <add hand="hand2">tri
    <layerStart layer="l2" id="ls1b" />cky!</add> boundary crossings
    to make the story <seg rend="90">com<layerEnd layer="l2" id="le1b" />plete.
    </seg></p>
    ...
  </body>
  <back>
    <join targets="ls1 le1 ls1b le1b" result="div" desc="physical layer" />
  </back>
</text>
```

Note Although the type attribute is not obligatory, DALF encoders are strongly encouraged to use it on each layer definition. Furthermore, although the global id attribute is never obligatory, it *must* be present on each **<layer>** element, in order to provide a reference point for the **<layerStart />** (p. 101) and **<layerEnd />** (p. 99) elements that signal the boundaries of those different physical layers in the document.

Content May contain additional paragraphs.

Parents layerList

Children p

Declaration

```
<!ELEMENT layer %om.RR;
        (p)*>
<!ATTLIST layer
        %a.global;
        type CDATA #IMPLIED>
```

See further 6. Correlations of logical and physical structures (p. 63)

A.1.a.30 <layerEnd />

Description Marks the end of a different physical layer in a letter.

Attributes In addition to global (p. 141) attributes:

layer Refers to a previously defined physical layer.

Datatype: IDREF

Values: Must be one of the physical layer identifiers declared in the header (see A.1.a.29 **<layer>** (p. 98)).

Default: #REQUIRED

Example

```
...
<layerList>
   <layer id="l2" type="sticker" />
</layerList>
...
<p>Gister-morgen heb ik de proef en den brief, om 8&frac12;u met de gewone post
ontvangen
en beide stukken met een etiket voorzien:<layerStart id="l1s" layer="l2" />
<print type="postal_tag">REMIS A LA POSTE<lb /> l' adresse &eacute;tant
inexacte<lb />
ou insuffisante.<lb /></print><layerEnd id="l1e" layer="l2" />
W&agrave;t er wel aan
de adressen mankeerde of te kort was, kan ik niet raden. ... </p>
```

Class Incl

Content This is an empty element.

Parents ab abbr actor add addrLine address addressee altName analytic app arg argument author authority back bibl biblFull biblScope biblStruct bloc body byline calc camera caption castGroup castItem castList cell cit classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor div div0 div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle edition editor emph envPart epigraph epilogue expan extent figDesc figure foreign front funder fw gloss group head headItem headLabel height hi imprimatur imprint institution item l label language lem lg list listBibl measure meeting mentioned metDecl monogr name note num opener oper orig p performance placeLet placeName principal print prologue ps pubPlace publicationStmt publisher q quote rdg rdgGrp ref reg region rendition repository resp respStmt restore result role roleDesc row rs salute seg series set settlement sic signed soCalled sound sp speaker sponsor stage street supplied symbol table tagUsage tech term text time timeRange title titlePage titlePart trailer unclear view width wit witDetail witList witness xref

Children EMPTY

Declaration

```
<!ELEMENT layerEnd %om.RR;
        EMPTY>
<!ATTLIST layerEnd
        %a.global;
        layer IDREF #REQUIRED>
```

See further 6. Correlations of logical and physical structures (p. 63)

A.1.a.31 <layerList>

Description Contains a definition of the different physical layers in the source document.

Attributes No other than global (p. 141) attributes

Example

```
<layerList>
   <layer id="l2" type="post-it" />
   <layer id="l3" type="sticker" />
   <layer id="l4" type="object" />
</layerList>
```

Content Must contain at least one list item.

Parents profileDesc

Children layer

Declaration

```
<!ELEMENT layerList %om.RR;
        (layer*)>
<!ATTLIST layerList
        %a.global;>
```

See further 6. Correlations of logical and physical structures (p. 63)

A.1.a.32 <layerStart />

Description Marks the start of a different physical layer in a letter.

Attributes In addition to global (p. 141) attributes:

layer Refers to a previously defined physical layer.

Datatype: IDREF

Values: Must be one of the physical layer identifiers declared in the header (see A.1.a.29 **<layer>** (p. 98)).

Default: #REQUIRED

Example

```
...
<layerList>
   <layer id="l2" type="sticker" />
</layerList>
...
<p>Gister-morgen heb ik de proef en den brief, om 8&frac12;u met de gewone post
ontvangen
en beide stukken met een etiket voorzien: <layerStart id="l1s" layer="l2" />
<print type="postal_tag">REMIS A LA POSTE<lb /> l' adresse &eacute;tant
inexacte<lb />
ou insuffisante.<lb /></print><layerEnd id="l1e" layer="l2" /> W&agrave;t er wel aan
de adressen mankeerde of te kort was, kan ik niet raden. ... </p>
```

Class Incl

Content This is an empty element.

Parents ab abbr actor add addrLine address addressee altName analytic app arg argument author authority back bibl biblFull biblScope biblStruct bloc body byline calc camera caption castGroup castItem castList cell cit classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor div div0 div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle edition editor emph envPart epigraph epilogue expan extent figDesc figure foreign front funder fw gloss group head headItem headLabel height hi imprimatur imprint institution item l label language lem lg list listBibl measure meeting mentioned metDecl monogr name note num opener oper orig p performance placeLet placeName principal print prologue ps pubPlace publicationStmt publisher q quote rdg rdgGrp ref reg region rendition repository resp respStmt restore result role roleDesc row rs salute seg series set settlement sic signed soCalled sound sp speaker sponsor stage street supplied symbol table tagUsage tech term text time timeRange title titlePage titlePart trailer unclear view width wit witDetail witList witness xref

Children EMPTY

Declaration

```
<!ELEMENT layerStart %om.RR;
        EMPTY>
```

```
<!ATTLIST layerStart
        %a.global;
        layer IDREF #REQUIRED>
```

See further 6. Correlations of logical and physical structures (p. 63)

A.1.a.33 <layout>

Description Contains a description of specific layout aspects of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<layout>
  <p>written in 3 columns, starting with the rightmost and continuing to the left</p>
</layout>
```

Content Must contain one or more paragraphs.

Parents physDesc

Children p

Declaration

```
<!ELEMENT layout %om.RR;
        (p+) >
<!ATTLIST layout
        %a.global;>
```

See further 4.3.4 Layout aspects: **<layout>** (p. 27)

A.1.a.34 <letContents>

Description Contains a description of the intellectual contents of the letter.

Attributes In addition to global (p. 141) attributes:

defective Indicates whether the contents of the letter being described is defective, i.e. incomplete.

Datatype: (yes | no | unk)

Legal values are:

'yes' the contents of the letter is incomplete
'no' the contents of the letter is complete
'unk' it is unknown whether the contents of the letter is

complete

Default: no

Example

```
<letContents>
  <class>love letter</class>
  <p>Streuvels proposes to his girlfriend</p>
</letContents>
```

Content Must contain one or more paragraphs; possibly preceded by a functional-communicative class indication of the letter. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children class note p

Declaration

```
<!ELEMENT letContents %om.RR;
        (class*, p+, note*)>
<!ATTLIST letContents
        %a.global;
        defective (yes|no|unk) "no">
```

See further 4.5 The contents of the letter: **<letContents>** (p. 34)

A.1.a.35 <letDesc>

Description Groups together all letter-specific metadata for a DALF document.

Attributes In addition to global (p. 141) attributes:

status Indicates the compositional status of a letter.

Datatype: (uni | compo | frag | def | unk)

Legal values are:

'uni'	the letter described is a complete entity which exists as a unitary unit
'compo'	the letter described is a complete entity comprising multiple units of different origin
'frag'	the letter described is a fragmentary entity, of which the majority is missing
'def'	the letter described is a defective unit, of which the majority is preserved

‘unk’ no claim can be made regarding the compositional status of a letter

Default: #IMPLIED

Example

```
<letDesc>
  <letIdentifier>
    <country>Belgium</country>
    <settlement>Antwerp</settlement>
    <repository>AMVC</repository>
    <collection>S 154/1235</collection>
    <idno>345/24</idno>
  </letIdentifier>
  <letHeading>
    <author>Stijn Streuvels</author>
    <addressee>Lannoo</addressee>
    <placeLet>Ingooigem</placeLet>
    <dateLet>1943-04-24</dateLet>
  </letHeading>
  <physDesc>
    <type>letter</type>
    <support>
      <p>single page with pre-printed letterhead, writing on both sides</p>
    </support>
    <extent>
      <dimensions>
        <height units="mm">214</height>
        <width units="mm">276</width>
      </dimensions>
    </extent>
  </physDesc>
  <envOcc occ="yes" />
  <letContents>
    <p>Joris Lannoo asks Streuvels what to do ...</p>
  </letContents>
</letDesc>
```

Note The **<letDesc>** element is the distinguishing feature of the DALF header, declared as an expansion of the standard TEI **[<sourceDesc>]** element.

Content Must contain exactly one of each of the mandatory child elements. The optional ones may occur only once and must occur in the order specified. **<letPart>** is the only optional element that may occur more than once. The note(s) may only be used to provide additional information after all other elements.

Parents sourceDesc

Children additional envOcc history letContents letHeading letIdentifier letPart note physDesc

Declaration

```
<!ELEMENT letDesc %om.RR;
        (letIdentifier, letHeading, physDesc, envOcc, letContents?, history?,
        additional?, letPart*, note*)>
<!ATTLIST letDesc
        %a.global;
        status (uni | compo | frag | def | unk) #IMPLIED>
```

See further 4. The DALF header (p. 11)

A.1.a.36 <letHeading>

Description Contains a structured description of bibliographical information of a letter.

Attributes No other than global (p. 141) attributes

Example

```
<letHeading>
  <author>Stijn Streuvels</author>
  <addressee>Maurice De Meyer</addressee>
  <placeLet>Ingooigem</placeLet>
  <dateLet>1945-01-13</dateLet>
</letHeading>
```

Content Must contain all of the mandatory child elements. When optional ones are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children addressee author dateLet note placeLet respStmt

Declaration

```
<!ELEMENT letHeading %om.RR;
        (author+, addressee+, respStmt*, placeLet, dateLet, note*)>
<!ATTLIST letHeading
        %a.global;>
```

See further 4.2 The letter heading: **<letHeading>** (p. 18)

A.1.a.37 <letIdentifier>

Description Contains information concerning the identification of the letter within its holding institution.

Attributes No other than global (p. 141) attributes

Example

```
<letIdentifier>
  <country>Belgium</country>
  <settlement>Antwerp</settlement>
  <repository>AMVC</repository>
  <collection>S 935/B2</collection>
  <idno>171373/2882</idno>
</letIdentifier>
```

Content Must contain all of the mandatory child elements. When optional ones are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc

Children altName collection country idno institution note region repository settlement

Declaration

```
<!ELEMENT letIdentifier %om.RR;
        (country, region?, settlement, institution?, repository, collection, idno, altName*,
note*)>
<!ATTLIST letIdentifier
        %a.global;>
```

See further 4.1 The letter identifier: **<letIdentifier>** (p. 14)

A.1.a.38 <letPart>

Description Contains metadata about distinct parts of a letter.

Attributes No other than global (p. 141) attributes

Example

```
<letDesc>
  <letIdentifier>
    <country>Belgium</country>
    <settlement>Antwerp</settlement>
    <repository>AMVC</repository>
    <collection>S 154/1235</collection>
    <idno>223949</idno>
  </letIdentifier>
  <letHeading>
    <author>Stijn Streuvels</author>
    <addressee>Lannoo</addressee>
    <placeLet>Ingooigem</placeLet>
    <dateLet>1943-04-24</dateLet>
  </letHeading>
```

```
<physDesc>
  <type>letter</type>
  <support>
    <p>single page with pre-printed letterhead, writing on both sides</p>
  </support>
  <extent>
    <dimensions>
      <height units="mm">214</height>
      <width units="mm">276</width>
    </dimensions>
  </extent>
</physDesc>
<envOcc occ="yes" />
<letPart>
  <letHeading>
    <author>Joris Lannoo</author>
    <addressee>Stijn Streuvels</addressee>
    <placeLet>Tielt</placeLet>
    <dateLet>1943-04-20</dateLet>
  </letHeading>
  <physDesc>
    <type>letter</type>
    <support>
      <p>typed letter on recto side of the letter</p>
    </support>
    <extent>
      <dimensions>
        <height units="mm">214</height>
        <width units="mm">276</width>
      </dimensions>
    </extent>
  </physDesc>
  <letContents>
    <p>Joris Lannoo asks Streuvels what to do ...</p>
  </letContents>
</letPart>
<letPart>
  <letHeading>
    <author>Joris Lannoo</author>
    <addressee>Stijn Streuvels</addressee>
    <placeLet>Tielt</placeLet>
    <dateLet>1943-04-20</dateLet>
  </letHeading>
  <physDesc>
    <type>letter</type>
    <support>
      <p>written letter on verso side of the letter</p>
    </support>
    <extent>
      <dimensions>
        <height units="mm">214</height>
        <width units="mm">276</width>
      </dimensions>
    </extent>
```

```
    </physDesc>
    <letContents>
      <p>Streuvels advises Lannoo to ...</p>
    </letContents>
  </letPart>
</letDesc>
```

Content Must contain at least one of the child elements; when more are used, they must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children additional history idno letContents letHeading letPart note physDesc

Declaration

```
<!ELEMENT letPart %om.RR;
        ((idno, letHeading?, physDesc?, letContents?, history?, additional?, letPart*,
note*) |
        (letHeading, physDesc?, letContents?, history?, additional?, letPart*, note*) |
        (physDesc, letContents?, history?, additional?, letPart*, note*) |
        (letContents, history?, additional?, letPart*, note*) |
        (history, additional?, letPart*, note*) |
        (additional, letPart*, note*) |
        (letPart+, note*))>
<!ATTLIST letPart
        %a.global;>
```

See further 4.8 Distinct letter parts: **<letPart>** (p. 43)

A.1.a.39 <musicNotation>

Description Describes the type of musical notation that appears in the letter.

Attributes No other than global (p. 141) attributes

Example

```
<musicNotation>
  <p>theme of "Struggle for Pleasure" in 3 staves</p>
  <p>theme of "The Scene" in 2 staves</p>
</musicNotation>
```

Content Must contain one or more paragraphs.

Parents physDesc

Children p

Declaration

```
<!ELEMENT musicNotation %om.RR;
```

```
        (p+) >
<!ATTLIST musicNotation
        %a.global;>
```

See further 4.3.5 Musical notation: **<musicNotation>** (p. 28)

A.1.a.40 <oper>

Description Contains the operator of a calculation.

Attributes No other than global (p. 141) attributes

Example

```
<calc>
  <arg>969 <abbr expan="exemplaren">ex.</abbr> (zie afrekening van
30.8.41)</arg>
  <oper>-</oper>
  <arg>138<abbr expan="exemplaren">ex</abbr> (
    <arg>133 <abbr expan="exemplaren">ex.</abbr> verkocht</arg>
    <oper>+</oper>
    <arg>5<abbr expan="persexemplaren">persex.</abbr></arg>)
  </arg>
  <result><hi rend="double_underlined">831</hi><abbr
expan="exemplaren">ex.</abbr></result>
</calc>
```

Content May contain character data and phrase-level elements.

Parents calc

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT oper %om.RR;
        (%phrase;)*>
<!ATTLIST oper
        %a.global;>
```

See further 5.3 Calculations: **<calc>** (p. 52)

A.1.a.41 <origin>

Description Contains any descriptive or other information concerning the origin of a letter.

Attributes No other than global (p. 141) attributes

Example

```
<origin>
   <p>draft of letter found, dated 17 August 1924</p>
</origin>
```

Note This element is reserved solely for the description of the origination process of the letter. It does not cover facts about its history before it entered the holding institution (see **<provenance>** (p. 121), the acquisition process by which the letter entered the holding institution (see acquisition (p. 74)), or its history after it entered the holding institution (see **<custodialHist>** (p. 83)).

The historical events should be listed in chronological order.

Content Must contain one or more paragraphs.

Parents history

Children p

Declaration

```
<!ELEMENT origin %om.RR;
        (p+)>
<!ATTLIST origin
        %a.global;>
```

See further 4.6 The history of the letter: **<history>** (p. 35)

A.1.a.42 <paraph />

Description Marks the occurrence of a parapharnalia, ie. a decorative element that has originated independently of the writing act.

Attributes In addition to global (p. 141) attributes:

paraphRef efers to a previously defined paraphernalia

Datatype: IDREF

Values: must be one of the paraphernalia identifiers declared in the header (see A.1.a.47 **<paraphernalia>** (p. 115))

Default: #REQUIRED

entity names the external entity within which the graphic image of the paraphernalia is stored.

Datatype: ENTITY

Default: #IMPLIED

Example

```
<?xml version="1.0"?>
<!DOCTYPE TEI.2 PUBLIC "//CTB//DTD Dalf 1.0 (based on TEI)//NL"
"DALF.dtd" [
<!NOTATION png PUBLIC
   '-//TEI//NOTATION IETF RFC2083 Portable Network Graphics//EN'>
<!ENTITY obj1 SYSTEM "obj1.png" NDATA png>
]>
<TEI.2>
  <text>
    ...
    <p>Dit bloempje vond ik onderweg: <paraph paraphRef="obj1"
entity="obj1"/>.</p>
    ...
  </text>
</TEI.2>
```

Class Incl

Content This is an empty element.

Parents ab abbr actor add addrLine address addressee altName analytic app arg argument author authority back bibl biblFull biblScope biblStruct bloc body byline calc camera caption castGroup castItem castList cell cit classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor div div0 div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle edition editor emph envPart epigraph epilogue expan extent figDesc figure foreign front funder fw gloss group head headItem headLabel height hi imprimatur imprint institution item l label language lem lg list listBibl measure meeting mentioned metDecl monogr name note num opener oper orig p performance placeLet placeName principal print prologue ps pubPlace publicationStmt publisher q quote rdg rdgGrp ref reg region rendition repository resp respStmt restore result role roleDesc row rs salute seg series set settlement sic signed soCalled sound sp speaker sponsor stage street supplied symbol table tagUsage tech term text time timeRange title titlePage titlePart trailer unclear view width wit witDetail witList witness

xref

Children EMPTY

Declaration

```
<!ELEMENT paraph %om.RR;
        EMPTY>
<!ATTLIST paraph
        %a.global;
        paraphRef IDREF #REQUIRED
        entity ENTITY #IMPLIED>
```

See further 5.5 Decorative elements: **<deco />** and **<paraph />** (p. 57)

A.1.a.43 <paraphDesc>

Description Contains a description of a paraphernalia in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<paraphDesc>
  <p>dried petal of a poppy</p>
</paraphDesc>
```

Content Must contain one or more paragraphs.

Parents paraphItem

Children p

Declaration

```
<!ELEMENT paraphDesc %om.RR;
        (p+)>
<!ATTLIST paraphDesc
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.44 <paraphItem>

Description Contains a structured description of a paraphernalia in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<paraphItem id="obj1">
  <paraphDesc>
```

```
    <p>dried petal of a poppy</p>
  </paraphDesc>
</paraphItem>
```

Note The **<paraphItem>** element is used in the header to define each paraphernalia that appears in the letter. One such element must appear within the header for each paraphernalia distinguished in the text.

Although the global id attribute is never obligatory, it *must* be present on each **<paraphItem>** element, in order to provide a reference point for the **<paraph />** (p. 111) element in the text that signals the occurrence of the paraphernalia.

Content Must contain at least a description of the paraphernalia, possibly followed by a transcription of text contained in the paraphernalia.

Parents paraphList

Children paraphDesc paraphText

Declaration

```
<!ELEMENT paraphItem %om.RR;
        (paraphDesc, paraphText?)>
<!ATTLIST paraphItem
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.45 <paraphList>

Description Contains a list of structured descriptions for each of the paraphernalia in a letter.

Attributes No other than global (p. 141) attributes

Example

```
<paraphList>
  <paraphItem id="obj1">
    <paraphDesc>
      <p>dried petal of a poppy</p>
    </paraphDesc>
  </paraphItem>
</paraphList>
```

Content Must contain at least one list item.

Parents paraphernalia

Children paraphItem

Declaration

```
<!ELEMENT paraphList %om.RR;
        (paraphItem+)>
<!ATTLIST paraphList
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.46 <paraphText>

Description Contains a rendering of text found as the contents of a decoration, respectively paraphernalia.

Attributes No other than global (p. 141) attributes

Example

```
<paraphText>
  <p>Brussel-Antwerpen <lb /> tweede klasse</p>
</paraphText>
```

Content Must contain one or more paragraphs.

Parents paraphItem

Children p

Declaration

```
<!ELEMENT paraphText %om.RR;
        (p+)>
<!ATTLIST paraphText
        %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.47 <paraphernalia>

Description Describes decorative items in the letter that have originated independently of the writing act.

Attributes No other than global (p. 141) attributes

Example

```
<paraphernalia>
  <paraphList>
    <paraphItem id="obj1">
      <paraphDesc>
```

```
        <p>dried petal of a poppy</p>
      </paraphDesc>
    </paraphItem>
  </paraphList>
</paraphernalia>
```

Example

```
<paraphernalia>
  <p>dried petal of a poppy</p>
</paraphernalia>
```

Note The paraphernalia in a document can be described either in a loose prose description, or in a list (recommended), containing a structured definition per paraphernalia concerned.

Content A prose description of the paraphernalia should contain one or more paragraphs; a structured description must be done in a single **<paraphList>** element.

Parents physDesc

Children p paraphList

Declaration

```
<!ELEMENT paraphernalia %om.RR;
      (paraphList | p+)>
<!ATTLIST paraphernalia
      %a.global;>
```

See further 4.3.6 Decorative elements: **<decoration>** and **<paraphernalia>** (p. 29)

A.1.a.48 <physDesc>

Description Contains a description of the physical appearance of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<physDesc>
  <type>letter</type>
  <support>
    <p>single page with pre-printed letterhead, writing on one side only</p>
  </support>
  <extent>
    <dimensions>
      <height units="mm">214</height>
      <width units="mm">276</width>
    </dimensions>
  </extent>
</physDesc>
```

Content Must contain exactly one of each of the mandatory child elements. When the optional ones are used, they may occur only once and must occur in the order specified. The note(s) may only be used to provide additional information after all other elements.

Parents letDesc letPart

Children condition decoration extent layout musicNotation note paraphernalia support type

Declaration

```
<!ELEMENT physDesc %om.RR;
        (type, support, extent, layout?, musicNotation?, decoration?, paraphernalia?,
condition?, note*)>
<!ATTLIST physDesc
        %a.global;>
```

See further 4.3 The physical description: **<physDesc>** (p. 24)

A.1.a.49 <placeLet>

Description Identifies the place where the letter was written.

Attributes In addition to global (p. 141) attributes and those inherited from names (p. ?) :

attested Indicates the status of the contents of the element on which it appears with regard to the evidence from which it is derived.

Datatype: (yes | no | added | unk)

Legal values are:

- **'yes'** the attribution in the element is made on the basis of evidence inside the letter
- **'added'** the attribution in the element is made on the basis of additional material accompanying the letter (eg. the envelope)
- **'no'** the attribution in the element is made on the basis of external evidence
- **'unk'** it is unknown on what basis the attribution in the element is made

Default: #IMPLIED

Example

```
<placeLet attested="added">Ingooigem</placeLet>
```

Class names

Content May contain character data and phrase-level elements.

Parents letHeading

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT placeLet %om.RR;
        (%phrase.seq;)>
<!ATTLIST placeLet
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED>
```

See further 4.2.2 The place and time of writing: **<placeLet>** and **<dateLet>** (p. 23)

A.1.a.50 <postmark>

Description Contains the postmark on an envelope.

Attributes No other than global (p. 141) attributes

Example

```
<postmark>
  <placeName>Tielt</placeName>
  <date value="1924-10-24">24.10.'24</date>
</postmark>
```

Content Must contain at least one of the child elements. The note(s) may only be used to provide additional information after all other elements.

Parents envPart

Children date figure note placeName

Declaration

```
<!ELEMENT postmark %om.RR;
        ((figure | placeName |date)+, note*)>
<!ATTLIST postmark
        %a.global;>
```

See further 5.1 The envelope: **<envelope>** (p. 47)

A.1.a.51 <print>

Description Contains printed material that was present on the carrier of the letter, or added afterwards.

Attributes In addition to global (p. 141) attributes:

style indicates recognised writing styles

Datatype: CDATA

Values: Any descriptive name such as ‘secretary’, ‘copperplate’, ‘Chancery’, ‘Italian’, etc.

Default: #IMPLIED

Note: This attribute is borrowed from the standard TEI element [handShift /] .

ink Describes tint or type of ink. May also be used to indicate the writing medium.

Datatype: CDATA

Values: Any descriptive name such as ‘ brown’, ‘pencil’, etc.

Default: #IMPLIED

Note: This attribute is borrowed from the standard TEI element [handShift /] .

character Describes other characteristics of the hand, particularly those related to the quality of the writing.

Datatype: CDATA

Values: Any descriptive name such as ‘shaky’, ‘thick’, ‘regular’.

Default: #IMPLIED

Note: This attribute is borrowed from the standard TEI

element [handShift /] .

hand Signifies the hand of the agent which made the addition.

Datatype: IDREF

Values: Must be one of the hand identifiers declared in the document header.

Default: %INHERITED;

Note: This attribute is borrowed from the standard TEI element [add] .

type Describes the type of printed material.

Datatype: CDATA

Values: Any convenient typology can be used. Sensible values may be ‘letterhead’, ‘stamp’, etc.

Default: #IMPLIED

Example

```
<print type="letterhead">FRANK&middot;LATEUR</print>
```

Class Incl

Content May contain character data and phrase-level elements.

Parents ab abbr actor add addrLine address addressee altName analytic app arg argument author authority back bibl biblFull biblScope biblStruct bloc body byline calc camera caption castGroup castItem castList cell cit classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor div div0 div1 div2 div3 div4 div5 div6 div7 docAuthor docDate docEdition docImprint docTitle edition editor emph envPart epigraph epilogue expan extent figDesc figure foreign front funder fw gloss group head headItem headLabel height hi imprimatur imprint institution item l label language lem lg list listBibl measure meeting mentioned metDecl monogr name note num opener oper orig p performance placeLet placeName principal print prologue ps pubPlace publicationStmt publisher q quote rdg rdgGrp ref reg region rendition repository resp respStmt restore result role roleDesc row rs salute seg series set

settlement sic signed soCalled sound sp speaker sponsor stage street supplied symbol table tagUsage tech term text time timeRange title titlePage titlePart trailer unclear view width wit witDetail witList witness xref

Children #PCDATA ab abbr add addSpan address alt altGrp anchor app bibl biblFull biblStruct calc camera caption castList cb cit corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp l label layerEnd layerStart lb lg link linkGrp list listBibl measure mentioned milestone move name note num orig p paraph pb placeName print ptr q quote ref reg restore rs seg sic soCalled sound sp space stage supplied table tech term text time timeRange timeline title unclear view witDetail witList xptr xref

Declaration

```
<!ELEMENT print %om.RR;
        %specialPara;>

<!ATTLIST print
        %a.global;
        style CDATA #IMPLIED
        ink CDATA #IMPLIED
        character CDATA #IMPLIED
        hand IDREF #IMPLIED
        type CDATA #IMPLIED>
```

See further 5.4 Pre- and post-printed material: **<print>** (p. 55)

A.1.a.52 <provenance>

Description Contains any descriptive or other information concerning the history of a letter, after its creation but before its acquisition.

Attributes No other than global (p. 141) attributes

Example

```
<provenance>
  <p>between January 1925 and 3 September 1935, the letter resided
  in the private collection of the receiver</p>
  <p>between the death of the receiver on 3 September 1935 and 7
  February 1937, the letter was owned by the heirs of the reveiver</p>
</provenance>
```

Note This element is reserved solely for the description of the history of the letter before it entered the holding institution. It does not cover facts about the origination of the letter (see **<origin>** (p. 110)), the acquisition process by which it entered the holding institution (see

acquisition (p. 74)), or its history after it entered the holding institution (see **<custodialHist>** (p. 83)).

The historical events should be listed in chronological order.

Content Must contain one or more paragraphs.

Parents history

Children p

Declaration

```
<!ELEMENT provenance %om.RR;
        (p+)>
<!ATTLIST provenance
        %a.global;>
```

See further 4.6 The history of the letter: **<history>** (p. 35)

A.1.a.53 <ps>

Description Contains a postscript in a letter or letter part.

Attributes No other than global (p. 141) attributes

Example

```
<closer>
  <salute>Met vriendelijken groet</salute>
  <signed>(Styn Streuvels)</signed>
</closer>
<ps>
  <p id="xr2"><add id="add1"><abbr expan="postscriptum">P.S.</abbr> Ze jubileeren
  bij de firma Veen (60 jaar bestaan)<ref target="n8">8</ref> en er wordt me daarom
  gevraagd, door het comit&eacute;: hoeveel geld ik daarvoor als feestgave wensch te
  geven! Zonderlinge zeden? Als ik nu eens vroeg: hoeveel ze voor mij beschikken als
  75-jarige jubilaris!</add></p>
</ps>
```

Class divbot

Content May contain character data and phrase-level elements.

Parents body div div0 div1 div2 div3 div4 div5 div6 div7 epilogue group lg performance prologue

Children #PCDATA ab abbr add addSpan address alt altGrp anchor app bibl biblFull biblStruct calc camera caption castList cb cit corr damage date dateRange deco del delSpan dimensions distinct emph expan figure

foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp l label layerEnd layerStart lb lg link linkGrp list listBibl measure mentioned milestone move name note num orig p paraph pb placeName print ptr q quote ref reg restore rs seg sic soCalled sound sp space stage supplied table tech term text time timeRange timeline title unclear view witDetail witList xptr xref

Declaration

```
<!ELEMENT ps %om.RR;
        %specialPara;>
<!ATTLIST ps
        %a.global;>
```

See further 5.2 The postscript: **<ps>** (p. 51)

A.1.a.54 <remarks>

Description Contains any comments or remarks not forming part of the description proper, for use by the cataloguer.

Attributes No other than global (p. 141) attributes

Example

```
<remarks>
  <p>maybe this letter can turn out significant for the retracing of the
  missing letters by Streuvels to Excelsior between 1926 and 1928</p>
</remarks>
```

Content Must contain one or more paragraphs.

Parents adminInfo

Children p

Declaration

```
<!ELEMENT remarks %om.RR;
        (p+)>
<!ATTLIST remarks
        %a.global;>
```

See further 4.7.1 Administrative information: **<adminInfo>** (p. 39)

A.1.a.55 <repository>

Description Contains the name of a repository (usually a distinct physical building) within which manuscripts are stored, forming part of an institution.

Attributes No other than global (p. 141) attributes and those inherited from names (p. 145) and typed (p. 148)

Example
```
<repository key="loc01_AMVC" reg="AMVC">AMVC</repository>
```

Class names; typed

Content May contain character data and phrase-level elements.

Parents letIdentifier

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration
```
<!ELEMENT repository %om.RR;
        (%phrase.seq;)>
<!ATTLIST repository
        %a.global;
        %a.names;
        %a.typed;>
```

See further 4.1.1 The macro-location path of a letter: **<country>**, **<region>**, **<settlement>**, **<institution>** and **<repository>** (p. 15)

A.1.a.56 <result>

Description Contains the result of a calculation.

Attributes No other than global (p. 141) attributes

Example
```
<calc>
  <arg>969 <abbr expan="exemplaren">ex.</abbr> (zie afrekening van
30.8.41)</arg>
  <oper>-</oper>
  <arg>138<abbr expan="exemplaren">ex</abbr> (
    <arg>133 <abbr expan="exemplaren">ex.</abbr> verkocht</arg>
    <oper>+</oper>
    <arg>5<abbr expan="persexemplaren">persex.</abbr></arg>)
  </arg>
  <result><hi rend="double_underlined">831</hi><abbr
expan="exemplaren">ex.</abbr></result>
```

```
</calc>
```

Note An argument can be defined as the right-hand side of an equation in a calculation. It can thus only occur as a descendant of a **<calc>** element. An argument may be in strictly numerical form, or may be intermingled with textual data.

Content May contain character data, phrase-level elements and embedded calculations and arguments.

Parents calc

Children #PCDATA abbr add addSpan address alt altGrp anchor app arg calc cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT result %om.RR;
        (%phrase; | calc | arg)*>
<!ATTLIST result
        %a.global;>
```

See further 5.3 Calculations: **<calc>** (p. 52)

A.1.a.57 <support>

Description Gives a description of the material on which the letter is written.

Attributes No other than global (p. 141) attributes

Example

```
<support>
  <p>very thin, semi-transparent paper</p>
</support>
```

Content Must contain one or more paragraphs.

Parents physDesc

Children p

Declaration

```
<!ELEMENT support %om.RR;
        (p+) >
```

```
<!ATTLIST support
        %a.global;>
```

See further 4.3.2 Support material: **<support>** (p. 26)

A.1.a.58 <surrogates>

Description Contains information about any other formal representations that exist of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<surrogates>
  <p>
    <bibl>
      digital facsimile, saved as JPEG image file
      <title type="filename">LS081141.jpg</title>
      <idno>AMVC 911/5 jpg</idno>
      <date>September 1996</date>
    </bibl>
  </p>
</surrogates>
```

Content Must contain one or more paragraphs.

Parents additional

Children p

Declaration

```
<!ELEMENT surrogates %om.RR;
        (p+)>
<!ATTLIST surrogates
        %a.global;>
```

See further 4.7.2 Other representations: **<surrogates>** (p. 41)

A.1.a.59 <type>

Description Gives a formal characterisation of the letter.

Attributes No other than global (p. 141) attributes

Example

```
<type>
  postcard
</type>
```

Content May contain character data only.

Parents physDesc

Children #PCDATA

Declaration

```
<!ELEMENT type %om.RR;
        (#PCDATA) >
<!ATTLIST type
        %a.global;>
```

See further 4.3.1 Formal characterisation: **<type>** (p. 26)

A.1.a.60 <width>

Description Contains a measurement for the width of an object.

Attributes No other than global (p. 141) attributes and those inherited from measured (p. 139)

Example

```
... het kaft is <dimensions>
   <width units="cm">25 cm</width>
   breed
</dimensions>
```

Class measured

Content May contain character data and phrase-level elements.

Parents dimensions

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT width %om.RR;
        (%phrase;)*>
<!ATTLIST width
        %a.global;
        %a.measured;>
```

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

Appendix A.1.b Modified TEI elements

A.1.b.1 <address>

Description Contains a postal or other address, for example of a publisher, an organisation, or an individual.

Attributes In addition to global (p. 141) attributes:

type Describes the type of address.

Datatype: CDATA

Values: Any convenient typology can be used.

Default: #IMPLIED

Example:

```
<address type="receiver">
  <addrLine>Stijn Streuvels</addrLine>
  <addrLine>Lijsternest</addrLine>
  <addrLine>Ingooigem</addrLine>
</address>
```

Note: Apart from the type attribute, the **<address>** element is taken over literally from the TEI scheme (see [the TEI P4 Guidelines]). This attribute is added in the DALF DTD to indicate sensible address types that can occur in letters, such as eg. ‘receiver’, ‘sender’, etc.

Example

```
<address type="receiver">
  <addrLine>Stijn Streuvels</addrLine>
  <addrLine>Lijsternest</addrLine>
  <addrLine>Ingooigem</addrLine>
</address>
```

Note Addresses may be encoded either as a sequence of lines, or using any sequence of address component elements.

Class data

Content If given as running prose, use a consistent format wherever possible,

for example separating lines of the address by commas, and including any postal code in the standard form.

Parents ab abbr actor add addrLine addressee altName arg author authority bibl biblScope bloc byline calc camera caption castItem catDesc cell classCode closer collection corr country creation damage date dateLet dateRange dateline del depth dimensions distance distinct distributor docAuthor docDate docEdition docImprint edition editor emph envPart expan extent figDesc foreign funder fw gloss head headItem headLabel height hi imprimatur institution item l label language lem measure meeting mentioned name note num opener oper orig p placeLet placeName principal print ps pubPlace publicationStmt publisher q quote rdg ref reg region rendition repository resp restore result role roleDesc rs salute seg settlement sic signed soCalled sound speaker sponsor stage street supplied symbol tagUsage tech term time timeRange title titlePart trailer unclear view width wit witDetail witness xref

Children add addSpan addrLine alt altGrp anchor cb deco delSpan figure fw gap index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp milestone name note paraph pb postBox postCode print seg street timeline

Declaration

```
<!ELEMENT address %om.RR;
        ((%m.Incl;)*, ((addrLine, (%m.Incl;)*)+ | ((%m.addrPart;), (%m.Incl;)*)* ))>
<!ATTLIST address
        %a.global;
        type CDATA #IMPLIED>
```

See further 5.1 The envelope: **<envelope>** (p. 47)

A.1.b.2 <author>

Description In a bibliographic reference, contains the name of the author(s), personal or corporate, of a work; the primary statement of responsibility for any bibliographic item.

Attributes In addition to global (p. 141) attributes and those inherited from names (p. ?) :

attested Indicates the status of the contents of the element on which it appears with regard to the evidence from which it is derived.

Datatype: (yes | added | no | unk)

Legal values are:

‘yes’ the attribution in the element is made on the basis of evidence inside the letter

‘added’ the attribution in the element is made on the basis of additional material accompanying the letter (eg. the envelope)

‘no’ the attribution in the element is made on the basis of external evidence

‘unk’ it is unknown on what basis the attribution in the element is made

Default: #IMPLIED

accepted Indicates whether the attribution made in the element is generally accepted. This is an optional attribute with one of following three values. Generally accepted attributions are encoded with value "yes"; attributions that are not generally accepted with "no"; when no claim can be made regarding this status, the value "unk" is used.

Datatype: (yes | no | unk)

Legal values are:

‘yes’ the attribution in the element is generally accepted

‘no’ the attribution in the element is not generally accepted

‘unk’ it is unknown whether the attribution in the element is generally accepted

Default: #IMPLIED

Example

```
<author attested="yes">Stijn Streuvels</author>
```

Note The **<author>** element is originally a standard TEI element (see [the TEI P4 Guidelines]), which has been enriched with the attributes from the names (p. 145) class, and the attested and accepted attributes. By this the encoder can provide enough additional information to identify the author of a letter (part) in **<letHeading>**.

Class names; biblPart

Content May contain character data and phrase-level elements.

Parents analytic bibl letHeading monogr titleStmt

Children #PCDATA abbr add addSpan address alt altGrp anchor app cb corr damage date dateRange deco del delSpan dimensions distinct emph expan figure foreign formula fw gap gloss handShift hi index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp measure mentioned milestone name note num orig paraph pb placeName print ptr ref reg restore rs seg sic soCalled space supplied term time timeRange timeline title unclear xptr xref

Declaration

```
<!ELEMENT author %om.RR;
        (%phrase.seq;)>
<!ATTLIST author
        %a.global;
        %a.names;
        attested (yes | added | no | unk) #IMPLIED
        accepted (yes | no | unk) #IMPLIED>
```

See further 4.2.1 The communicative participants: **<author>**, **<addressee>** and **<respStmt>** (p. 20)

A.1.b.3 <biblStruct>

Description Contains a structured bibliographic citation, in which only bibliographic subelements appear and in a specified order.

Attributes No other than global (p. 141) attributes and those inherited from [bibl] and [declarable]

Example

```
<biblStruct>
  <monogr>
    <author>Streuvels, Stijn</author>
    <title>De teleurgang van den Waterhoek</title>
    <edition>eerste druk</edition>
    <imprint>
      <publisher>Excelsior</publisher>
      <pubPlace>Brugge</pubPlace>
      <publisher>Veen</publisher>
      <pubPlace>Amsterdam</pubPlace>
      <date>1927</date>
    </imprint>
  </monogr>
</biblStruct>
```

Note This element is taken over from the TEI scheme (see [the TEI P4 Guidelines]). However, its content model has been redefined without

the **<note>** element, to avoid interference with the declaration of **<note>** as a global element.

Class bibl; declarable

Parents ab add argument body camera caption castList cell cit corr country damage div div0 div1 div2 div3 div4 div5 div6 div7 docEdition emph epigraph epilogue figDesc foreign head hi imprimatur item l lem listBibl meeting metDecl note p performance print prologue ps q quote rdg ref region rendition seg set sic sound sourceDesc stage supplied tagUsage taxonomy tech title titlePart unclear view wit witDetail witness xref

Children add addSpan alt altGrp analytic anchor cb deco delSpan figure fw gap idno index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp milestone monogr note paraph pb print seg series timeline

Declaration

```
<!ELEMENT biblStruct %om.RO;
        ((%m.Incl;)*, (analytic, (%m.Incl;)*)?,
        ((monogr, (%m.Incl;)*), (series, (%m.Incl;)*)*)+,
        (idno, (%m.Incl;)*)*)>
<!ATTLIST biblStruct
        %a.global;
        %a.declarable;>
```

See further 7.3 Changes relating to the declaration of **<note>** as global element (p. 70)

A.1.b.4 <monogr>

Description Contains bibliographic elements describing an item (e.g. a book or journal) published as an independent item (i.e. as a separate physical object).

Attributes No other than global (p. 141) attributes and those inherited from [biblPart]

Example

```
<monogr>
  <author>Streuvels, Stijn</author>
  <title>De teleurgang van den Waterhoek</title>
  <edition>eerste druk</edition>
  <imprint>
    <publisher>Excelsior</publisher>
    <pubPlace>Brugge</pubPlace>
    <publisher>Veen</publisher>
    <pubPlace>Amsterdam</pubPlace>
    <date>1927</date>
  </imprint>
```

```
</monogr>
```

Note The **<monogr>** element may occur only within bibliographic citation or reference elements; it is mandatory for description of the monographic level of **<biblStruct>** elements.

This element is taken over from the TEI scheme (see [the TEI P4 Guidelines]). However, its content model has been redefined without the **<note>** element, to avoid interference with the declaration of **<note>** as a global element.

Class biblPart

Content May contain specialized bibliographic elements, in a prescribed order.

Parents bibl biblStruct

Children add addSpan alt altGrp anchor author biblScope cb deco delSpan edition editor extent figure fw gap imprint index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp meeting milestone note paraph pb print respStmt seg timeline title

Declaration

```
<!ELEMENT monogr %om.RO;
        (( (%m.Incl;)*,
            ((
              (author | editor | respStmt),
              (author | editor | respStmt | %m.Incl;)*,
              (title, (%m.Incl;)*)+,
              ( (editor | respStmt), (%m.Incl;)* )* )
            |(
              (title, (%m.Incl;)*)+,
              ((author | editor | respStmt), (%m.Incl;)*)*
            )))?,
            ((meeting), (%m.Incl;)*)*,
            (edition, (editor | respStmt | %m.Incl;)*)*, imprint,
            (imprint | extent | biblScope | %m.Incl;)*
           )>
<!ATTLIST monogr
        %a.global;>
```

See further 7.3 Changes relating to the declaration of **<note>** as global element (p. 70)

A.1.b.5 <profileDesc>

Description provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in

which it was produced, the participants and their setting.

Attributes No other than global (p. 141) attributes

Example

```
<profileDesc>
  <handList>
    <hand id="hand2" />
  </handList>
  <layerList>
    <layer id="l2" type="post-it" />
  </layerList>
</profileDesc>
```

Note This element is taken over literally from the TEI scheme (see [the TEI P4 Guidelines]), and extended with the **<layerList>** DALF element.

Parents teiHeader

Children creation handList langUsage layerList textClass

Declaration

```
<!ELEMENT profileDesc %om.RR;
        (creation?, langUsage*, textDesc*,particDesc*, settingDesc*,
        handList*, layerList*, textClass*)>
<!ATTLIST profileDesc
        %a.global;>
```

See further 6. Correlations of logical and physical structures (p. 63)

A.1.b.6 <sp>

Description Contains an individual speech in a performance text, or a passage presented as such in a prose or verse text.

Attributes In addition to global (p. 141) attributes and those inherited from [chunk] :

who identifies the speaker of the part by supplying an IDREF value.

Datatype: IDREFS

Values: The values used are derived from the id attribute on the **<role>** elements in the cast list or from a list of the participants.

Default: #IMPLIED

Example

```
<sp>
   <speaker>The reverend Doctor Opimiam</speaker>
   <p>I do not think I have named a single unpresentable fish.</p>
</sp>
<sp>
   <speaker>Mr Gryll</speaker>
   <p>Bream, Doctor: there is not much to be said for bream.</p>
</sp>
<sp>
   <speaker>The Reverend Doctor Opimiam</speaker>
   <p>On the contrary, sir, I think there is much to be said for him.
In the first place....</p>
   <p>Fish, Miss Gryll &mdash; I could discourse to you on fish by the hour:
but for the present I will forbear...</p>
</sp>
```

Note

The who attribute on this element may be used either in addition to the **<speaker>** element or as an alternative.

This element is taken over from the TEI scheme (see [the TEI P4 Guidelines]). However, its content model has been redefined without the **<seg>** element, to avoid interference with the declaration of **<seg>** as a global element.

Class

chunk

Content

Lines or paragraphs, stage directions, and phrase-level elements.

Parents

add argument body castList corr div div0 div1 div2 div3 div4 div5 div6 div7 epigraph epilogue item metDecl note performance print prologue ps q quote respStmt set sic stage view

Children

ab add addSpan alt altGrp anchor cb deco delSpan figure fw gap index interp interpGrp join joinGrp l layerEnd layerStart lb lg link linkGrp milestone note p paraph pb print seg speaker stage timeline

Declaration

```
<!ELEMENT sp %om.RO;
        ((%m.Incl;)*, (speaker, (%m.Incl;)*)?,((p | l | lg | ab | stage), (%m.Incl;)*)+)>
<!ATTLIST sp
        %a.global;
        who IDREFS #IMPLIED>
```

See further

7.4 Changes relating to the declaration of **<seg>** as global element (p. 72)

A.1.b.7 <sourceDesc>

Description Supplies a bibliographic description of the letter from which an electronic text was derived or generated. Specifically for the DALF DTD, it has been extended with an element that groups all letter-specific meta-information.

Attributes No other than global (p. 141) attributes and those inherited from [declarable]

Example

```
<sourceDesc>
  <letDesc>
    <letIdentifier>
      <country>Belgium</country>
      <settlement>Antwerp</settlement>
      <repository>AMVC</repository>
      <idno>S 154/1235</idno>
    </letIdentifier>
    <letHeading>
      <author>Stijn Streuvels</author>
      <addressee>Lannoo</addressee>
      <placeLet>Ingooigem</placeLet>
      <dateLet>1943-04-24</dateLet>
    </letHeading>
    <physDesc>
      <type>letter</type>
      <support>
        <p>single page with pre-printed letterhead, writing on both sides</p>
      </support>
      <extent>
        <dimensions>
          <height units="mm">214</height>
          <width units="mm">276</width>
        </dimensions>
      </extent>
    </physDesc>
    <envOcc occ="yes" />
    <letContents>
      <p>Joris Lannoo asks Streuvels what to do ...</p>
    </letContents>
  </letDesc>
</sourceDesc>
```

Note This element is taken over literally from the TEI scheme (see [the TEI P4 Guidelines]), and extended with the **<letDesc>** DALF element.

Class declarable

Content Must contain one letter description.

Parents biblFull fileDesc

Children biblStruct letDesc note

Declaration

```
<!ELEMENT sourceDesc %om.RR;
        (biblStruct?, letDesc, note*)>
<!ATTLIST sourceDesc
        %a.global;
        %a.declarable;>
```

See further 4. The DALF header (p. 11)

A.1.b.8 <text>

Description Contains a single letter, whether unitary or composite. Specifically for the DALF DTD, it has been extended with an element to indicate envelopes.

Attributes No other than global (p. 141) attributes and those inherited from [declaring] and inter (p. 145)

Example

```
<TEI.2>
  <teiHeader>...</teiHeader>
  <text>
    <envelope>...</envelope>
    <body>
      <opener>
        <address>...</address>
        <dateline>...</dateline>
        <salute>...</salute>
        ...
      </opener>
      <p>...</p>
      <closer>
        <salute>...</salute>
        <signed>...</signed>
        ...
      </closer>
      <ps>...</ps>
    </body>
    <back>
      <note>...</note>
      <join>...</join>
      ...
    </back>
  </text>
</TEI.2>
```

Note This element is taken over literally from the TEI scheme (see [the TEI P4 Guidelines]), and extended with the **<envelope>** (p. 95) DALF

element.

Class declaring; inter

Parents TEI.2 ab add camera caption cell corr country damage docEdition emph figDesc figure foreign group head hi imprimatur item l lem meeting note p print ps q quote rdg ref region rendition seg sic sound stage supplied tagUsage tech title titlePart unclear view wit witDetail witness xref

Children add addSpan alt altGrp anchor back body cb deco delSpan envelope figure front fw gap group index interp interpGrp join joinGrp layerEnd layerStart lb link linkGrp milestone note paraph pb print seg timeline

Declaration

```
<!ELEMENT text %om.RR;
        ((envelope | %m.Incl;)*,
        (front, (envelope | %m.Incl;)*)?,
        (body | group), (envelope | %m.Incl;)*,
        (back, (envelope | %m.Incl;)*)?)>
<!ATTLIST text
        %a.global;
        %a.declaring;>
```

See further 5.1 The envelope: **<envelope>** (p. 47)

Appendix A.2 Element and attribute classes

Appendix A.2.a DALF element and attribute classes

A.2.a.1 measured

Declaration

```
<!ENTITY % a.measured 'units CDATA #IMPLIED'>
```

Description Attributes which can be used to specify the measurement unit of the **<dimensions>** element and its child elements.

Attributes

units Specifies the units used for the measurement.

Datatype: CDATA

Values: Sensible values are abbreviations for measurement units like ‘cm’, ‘mm’, ‘inch’.

Default: #IMPLIED

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

Appendix A.2.b Modified TEI element and attribute classes

A.2.b.1 biblPart

Description Groups elements which can appear only within bibliographic citation elements.

Declaration

```
<!ENTITY % m.biblPart "analytic | author | biblScope | edition | editor | extent |
        idno | imprint | monogr | pubPlace | publisher | respStmt | series">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines])that has been redefined without the **<note>** element, to avoid interference with the declaration of **<note>** as a global element.

See further 7.3 Changes relating to the declaration of **<note>** as global element (p. 70)

A.2.b.2 common

Description Groups common chunk- and inter-level elements.

Declaration

```
<!ENTITY % x.common 'calc |'>
<!ENTITY % m.common " calc | bibl | biblFull | biblStruct | ab | eTree | graph | l | lg | p | sp |
        tree | witList | cit | q | quote | label | list | listBibl  | witDetail | stage | table">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the DALF element **<calc>** (p. 79) , and redefined without the **<figure>** element, to avoid interference with the declaration of **<figure>** as a global element.

See further 5.3 Calculations: **<calc>** (p. 52) ; 7.2 Changes relating to the declaration of **<figure>** as global element (p. 70)

A.2.b.3 divbot

Declaration

```
<!ENTITY % x.divbot 'ps |'>
<!ENTITY % m.divbot "%x.divbot; %n.byline; | %n.closer; | %n.dateline; | %n.epigraph; | %n.salute; | %n.signed; | %n.trailer;">
```

Description Groups elements which can occur at the end of a text division.

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the DALF element **<ps>** (p. 122) .

See further 5.2 The postscript: **<ps>** (p. 51)

A.2.b.4 edit

Description Groups phrase-level elements for simple editorial correction and transcription.

Declaration

```
<!ENTITY % m.edit "app | corr | damage | del | orig | reg | restore | sic | space | supplied | unclear">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]) that has been redefined without the **<add>** element, to avoid interference with the declaration of **<add>** as a global element.

See further 7.1 Changes relating to the declaration of **<add>** as global element (p. 69)

A.2.b.5 global

Description Defines a set of attributes common to all elements in the DALF encoding scheme.

Declaration

```
<!ENTITY % a.global '
        %a.terminology;
        %a.linking;
        %a.analysis;
        id ID #IMPLIED
        n CDATA #IMPLIED
        lang IDREF %INHERITED;
        rend CDATA #IMPLIED'>
```

Attributes From the [core TEI tagset] :

id Provides a unique identifier for the element bearing the ID value.

Datatype: CDATA

Values: Any valid name. Values must be unique in the document and start with a letter or the underscore character ("_"), and contain no characters other than letters, digits, hyphens, underscores, full stops, and certain combining and extension characters.

Default: #IMPLIED

n Gives a number (or other label) for an element, which is not necessarily unique within the document.

Datatype: CDATA

Values: Any string of characters; often, but not necessarily, numeric.

Default: #IMPLIED

lang Indicates the language of the element content, usually using a two- or three-letter code from ISO 639.

Datatype: IDREF

Values: Must be an identifier of a **<language>** element

supplied in the header (see [the TEI P4 Guidelines]).

Default: %INHERITED;

Note: If no value is specified for lang, the lang value for the immediately enclosing element is inherited; for this reason, a value should always be specified on the outermost element([**<TEI.2>**]).

rend Indicates how the element in question was rendered or presented in the source text.

Datatype: CDATA

Values: The encoder can develop any convenient typology for the indication of presentational aspects.

Default: #IMPLIED

From the additional tag set for [linking and segmentation] :

corresp Points to elements that correspond to the current element in some way.

Datatype: IDREFS

Values: Must be a space-separated list of one or more valid identifiers of other elements in the document.

Default: #IMPLIED

synch Points to elements that are synchronous with the current element.

Datatype: IDREFS

Values: Must be a space-separated list of one or more valid identifiers of other elements in the document.

Default: #IMPLIED

sameAs Points to an element that is the same as the current element.

Datatype: IDREF

Values: Must be a valid identifier of another element in the document.

Default: #IMPLIED

copyOf Points to an element of which the current element is a copy.

Datatype: IDREF

Values: Must be a valid identifier of another element in the document.

Default: #IMPLIED

next Points to the next element of a virtual aggregate of which the current element is part.

Datatype: IDREF

Values: Must be a valid identifier of another element in the document.

Default: #IMPLIED

prev Points to the previous element of a virtual aggregate of which the current element is part.

Datatype: IDREF

Values: Must be a valid identifier of another element in the document.

Default: #IMPLIED

exclude Points to elements that are in exclusive alternation with the current element.

Datatype: IDREFS

Values: Must be a space-separated list of one or more valid identifiers of other elements in the document.

Default: #IMPLIED

select Selects one or more alternants; if one alternant is selected,

the ambiguity or uncertainty is marked as resolved. If more than one alternant is selected, the degree of ambiguity or uncertainty is marked as reduced by the number of alternants not selected.

Datatype: IDREFS

Values: Must be a space-separated list of one or more valid identifiers of other elements in the document.

Default: #IMPLIED

from the additional tag set for [simple analysis] :

ana Indicates one or more elements containing interpretations of the element on which the ana attribute appears.

Datatype: IDREFS

Values: Must be a space-separated list of one or more valid identifiers of other elements in the document.

Default: #IMPLIED

See further 3.3 Global attributes (p. 8)

A.2.b.6 Incl

Description Groups empty elements which may appear at any point within a DALF text.

Declaration

```
<!ENTITY % x.Incl 'add | deco | figure | paraph | print | layerStart | layerEnd | note | seg |'>
<!ENTITY % m.Incl "%x.Incl; %n.anchor; | %m.editIncl; | %m.metadata; | %m.refsys;">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the TEI elements **[<add>]** , **[<figure>]** , **[<note>]** and **[<seg>]** , and the DALF elements **<deco />** (p. 85) , **<layerEnd />** (p. 99) , **<layerStart />** (p. 101) , **<paraph />** (p. 111) and **<print>** (p. 119) .

See further 5.4 Pre- and post-printed material: **<print>** (p. 55) ; 5.5 Decorative elements: **<deco />** and **<paraph />** (p. 57) ; 6. Correlations of logical and physical structures (p. 63) ; 7. Modifications to TEI element classes (p. 69)

A.2.b.7 inter

Description Groups elements of the intermediate (inter-level) class: these elements can occur both within and and between paragraphs or other chunk-level elements.

Declaration

```
<!ENTITY % x.inter 'calc |'>
<!ENTITY % m.inter "calc | bibl | biblFull | biblStruct | castList | cit | q | quote |
        label | list | listBibl | witDetail | stage | camera | caption | move | sound |
        tech | view | table | text">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the DALF element **<calc>** (p. 79) and redefined without the **<figure>** element, to avoid interference with the declaration of **<figure>** as a global element.

See further 5.3 Calculations: **<calc>** (p. 52) ; 7.2 Changes relating to the declaration of **<figure>** as global element (p. 70)

A.2.b.8 names

Description Groups those elements which refer to named persons, places, organisations etc.

Declaration

```
<!ENTITY % a.names '
        key CDATA #IMPLIED
        reg CDATA #IMPLIED'>
```

Attributes

key Provides an alternative identifier for the object being named, such as a database record key.

Datatype: CDATA

Values: Any string of characters.

Default: #IMPLIED

Note: The value may be a unique identifier from a database, or simply a more explicit name for the referent. Its purpose is only to record an identification.

reg Gives a normalised or regularised form of the name used.

Datatype: CDATA

Values: Any string of characters.

Default: #IMPLIED

Note: In providing a ‘regularised’ form, no claim is made that the form in the source text is incorrect; the regularised form is simply that chosen as the main form for purposes of unifying variant forms under a single heading.

Note This is a standard TEI attribute class (see [the TEI P4 Guidelines]), extended with the DALF elements **<addressee>** (p. 75) , **<altName>** (p. 78) , **<author>** (p. 129) , **<collection>** (p. 81) , **<institution>** (p. 97) , **<placeLet>** (p. 117) and **<repository>** (p. 123) .

See further 4.1.1 The macro-location path of a letter: **<country>**, **<region>**, **<settlement>**, **<institution>** and **<repository>** (p. 15) ; 4.1.2 The micro-location path of a letter: **<collection>**, **<idno>** and **<altName>** (p. 17) ; 4.2.1 The communicative participants: **<author>**, **<addressee>** and **<respStmt>** (p. 20) ; 4.2.2 The place and time of writing: **<placeLet>** and **<dateLet>** (p. 23)

A.2.b.9 notes

Description Groups all note-like elements.

Declaration

```
<!ENTITY % m.notes "witDetail">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]) that has been redefined without the **<note>** element, to avoid interference with the declaration of **<note>** as a global element.

See further 7.3 Changes relating to the declaration of **<note>** as global element (p. 70)

A.2.b.10 seg

Description Groups elements used for arbitrary segmentation.

Declaration
```
<!ENTITY % m.seg "c | cl | m | phr | s | w">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines] that has been redefined without the **<seg>** element, to avoid interference with the declaration of **<seg>** as a global element.

See further 7.4 Changes relating to the declaration of **<seg>** as global element (p. 72)

A.2.b.11 phrase

Description Groups those elements which can occur at the level of individual words or phrases.

Declaration
```
<!ENTITY % x.phrase 'dimensions |'>
<!ENTITY % m.phrase "%x.phrase; %m.data; | %m.edit; | %m.formPointers; |
        %n.formula; | %n.handShift; | %m.hqphrase; | %m.loc; | %m.phrase.verse; |
        %m.seg; | %m.sgmlKeywords;">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the DALF element **<dimensions>** (p. 91) .

See further 5.6 Detailed dimensions: **<dimensions>** (p. 59)

A.2.b.12 tpParts

Description Groups those elements which can occur as direct constituents of a title page.

Declaration
```
<!ENTITY % x.tpParts 'calc |'>
<!ENTITY % m.tpParts "%x.tpParts; %n.byline; | %n.docAuthor; | %n.docDate; |
        %n.docEdition; | %n.docImprint; | %n.docTitle; | %n.epigraph; |
        %n.figure; | %n.imprimatur; | %n.titlePart;">
```

Note This is a standard TEI element class (see [the TEI P4 Guidelines]), extended with the DALF element **<calc>** (p. 79) and redefined without the **<figure>** element, to avoid interference with the declaration of

<figure> as a global element.

See further 5.3 Calculations: **<calc>** (p. 52) ; 7.2 Changes relating to the declaration of **<figure>** as global element (p. 70)

A.2.b.13 typed

Description Defines a set of attributes which can be used to classify or subclassify certain elements in any way.

Declaration

```
<!ENTITY % a.typed '
        type CDATA #IMPLIED
        subtype CDATA #IMPLIED'>
```

Attributes

type Characterises the element in some sense, using any convenient classification scheme or typology.

Datatype: CDATA

Values: Any string of characters.

Default: #IMPLIED

subtype Provides a sub-categorisation of the element, if needed.

Datatype: CDATA

Values: Any string of characters.

Default: #IMPLIED

Note This is a standard TEI attribute class (see [the TEI P4 Guidelines]), extended with the DALF elements **<altName>** (p. 78) , **<collection>** (p. 81) , **<institution>** (p. 97) and **<repository>** (p. 123) .

See further 4.1.1 The macro-location path of a letter: **<country>**, **<region>**, **<settlement>**, **<institution>** and **<repository>** (p. 15) ; 4.1.2 The micro-location path of a letter: **<collection>**, **<idno>** and **<altName>** (p. 17)

Appendix B. The DALF DTD files

The files used to declare the DALF extensions and modifications, are

- [DALFExtns.ent]
- [DALFExtns.dtd]

Together they produced the fully expanded [DALF.dtd] file.