

Weed Markup Language (WeedML) Design

Version: 18 July 2007.

Abstract

The Weed Markup Language (WeedML) is an open, XML-based format for representing models of the population dynamics of agricultural weeds. WeedML is a software-independent language freely available on Internet. WeedML makes it easy to employ parameters defined in the Weed Traits Database (WTDB). WeedML is managed by the ENDURE RA4.5 Working Group but is open for research purposes for the whole scientific community.

1. Introduction

A WeedML file contains a static description of a weed population dynamics model. Only dedicated software, e.g. a *WeedML Simulator* or *WeedML Editor*, will make the description come alive, i.e. carry out a simulation run or provide editing facilities for the file.

WeedML is a dialect of XML, i.e. a WeedML file is also a valid XML file. Numerous books and web sites provides introductory material on XML. The official XML site can be found at www.w3.org/XML.

This document should, however, by examples give all readers a feeling for how WeedML files look, and how they are interpreted. An XML file, and hence a WeedML file, can be constructed with any simple text editor, e.g. Windows Notepad. A dedicated WeedML Editor will be developed to make model construction much easier.

2. Basics

A model defined in WeedML file consists of a collection of submodels.

If the Weed Traits Database is used to lookup parameter values, its location must be defined.

To define a concise vocabulary of weeds, crops and management practices, specific tables are defined.

For example:

```
<model name="Reduced tillage" based_upon="url(default_model.xml)/model"> (1)
  <database name="wtodb" location="url(312.419.213.445/wtodb)"/> (2)
  <table name="weeds"
    based_upon="url(default_tables.xml)/models/table[@name='weeds']"/> (3)
  <table name="crops"
    based_upon="url(default_tables.xml)/models/table[@name='crops']"/>
  <table name="management"
    based_upon="url(default_tables.xml)/models/table[@name='management']"/>
  <submodel type="weed" name="Polygonum sp" (4)
    based_upon="url(weed_models.xml)/model/weed[@name='summer annual']">
    <submodel type="seedbank"> (5)
      Details of submodel...
    </submodel>
    <submodel type="germination" method="germination calendar"> (6)
      resolution="monthly">
        Details of submodel...
      </submodel>
    <submodel type="plant growth" method="Holst 2008">
      <parameter name="juvenile duration"
        average="lookup(wtodb/polav/juvdur/avg)"/> (7)
      Details of submodel...
    </submodel>
    <submodel type="reproduction" method="biomass allometry">
      <submodel name="fecundity" type="power">
        <parameter name="a" avg="205.7" source="Collings et al. (2005)"/> (8)
        <parameter name="b" avg="1.0" source="Collings et al. (2005)"/>
      </submodel>
```

```

    </submodel>
</submodel>
<submodel type="weed" name="Papaver rhoeas"
    based_upon="url(weed_models.xml)/model/weed[@name=winter annual']">
    Details of submodel...
</submodel>
<submodel type="environment">
    Details of submodel...
</submodel>
    More submodels...
</model>

```

(9)

(1) A general feature of model elements is that they can be based upon an element defined in the same or another file. The derived element inherits all the contents of the base elements but is allowed to add or override contents based upon that. This allows the reuse of parameter values and of whole model structures between models.

(2) Here, we define a shorthand name “wtddb” for the Weed Traits Database residing somewhere on the Internet.

(3) A table containing weed names is defined based upon a table found in another file.

(4) There are many kinds of submodels. Here we define one of the type “weed” and give it a describing name “Polygonum sp”. It is based upon a general summer annual weed submodel residing in another WeedML file on the user’s computer.

(5) Submodels, like here “seed bank”, may be nested inside another submodel.

(6) If alternative version of a submodel exists, one must specify not only its type (here “germination”) but also the method by which it is implemented (here “germination calendar”). Some methods may demand additional information, in this case a time resolution given as “monthly”.

(7) At the most detailed level of the WeedML file, the values of model parameters are defined. In this case the value is looked up in the WTDB (Note: The specific syntax of this will be settled together with the final technical design of WTDB).

(8) Parameter values can also be given directly in the WeedML file.

(9) Weeds are not the only type of submodels. Section 3 gives more examples of WeedML submodels.

2.1 FILE PROCESSING

The order of submodels in a WeedML file has no consequence. As software reads the file, it builds an internal representation of the submodels, their functionality and interrelations.

2.2 CONCEPTS

The root element of a WeedML file must be a “model” element or “scenarios” element.

A scenarios element contains a collection of models. – Scenarios are meant to be an easy way of setting up a suite of little variations over the same model. They will not be further defined in this document.

A model contains a collection of submodels. In addition it may hold global definitions, such as a reference to the Weed Traits Database and tables with vocabularies of weeds, crops and management submodels.

Submodels may contain other submodels and parameters.

Submodels are classified by their ‘type’ and may have an identifying ‘name’. A ‘method’ may subclass a submodel by referring to a particular method of implementation.

2.3 DATA TYPES

A parameter is defined by its name and value set. The value set can contain values that are numerical (2.3.1), text (2.3.2) or lists (2.3.3), or which refers to a value set defined elsewhere, either in a WeedML file (2.3.4) or the Weed Traits Database (2.3.5).

Parameters may optionally have a 'units' attribute. This is useful, e.g. for defining the type of time scale implied for the duration of a life stage.

Other attributes can be added. The meaning of these is defined by the implementation of the software reading the WeedML file.

2.3.1 Number

A numerical parameter is defined by its statistics, such as average and variance, e.g.

```
<parameter name="a" average="205.7" variance="15.6"/>
```

Percentages can be written directly, e.g.

```
<parameter name="survival" average="45.6%"/>
```

2.3.2 Text

A parameter on nominal scale is defined by a value which, depending on the parameter, must be picked from a list of possible values, e.g. (harrowing, ploughing, ridging):

```
<parameter name="tillage" value="harrowing" depth="10"/>
```

In this example the tillage parameter also expects a depth attribute in the value set.

2.3.3 Lists

List of numbers and text can be defined, separated by spaces and nested by parentheses. E.g.,

```
<parameter name="mortality" value="((early mid late)(50% 60%)(0 0)(20% 30%))"/>
```

2.3.4 WeedML Link

The value set can be defined by reference to an existing value set, either in the same or another WeedML file:

```
<parameter name="seed halflife"
lookup="url(weed_models.xml)/model/submodel[@name='Poa annua']//
parameter[@name='seed halflife']"/>
```

The looked-up value set could itself refer further via yet another link.

Links follow the XLink standard (www.w3.org/TR/xlink/). Part of this is the XPath language which specifies how to navigate inside an XML document (www.w3.org/TR/xpath).

2.3.5 Weed Traits Database (WTDB) Link

A parameter value can be looked up in the Weed Traits Database:

```
<parameter name="juvenile duration" average="lookup(wtdb/polav/juvdur/avg)"
variance="lookup(wtdb/polav/juvdur/var)"/>
```

The details of this syntax awaits the final design of WTDB.

2.4 DEFAULTING

To facilitate reuse, any model, scenarios or submodel element may be based upon another model, scenario or submodel. This means that as a first step, all elements and attributes of the base element is copied to the new element. In a second step, the elements and attributes defined explicitly in the new element is added, overriding any definitions that were already present in the base element.

For example, in file basics.xml:

```
<model name="weed models">
  <submodel type="weed" name="Poa annua">
    <submodel type="plant growth" method="Holst 2008">
      <parameter name="juvenile duration"
        average="lookup(wtddb/polav/juvdur/avg)"/>
      Details of submodel...
    </submodel>
  </submodel>
  <submodel type="weed" name="Viola arvensis">
    Details of submodel...
  </submodel>
</model>
```

In file newmodel.xml:

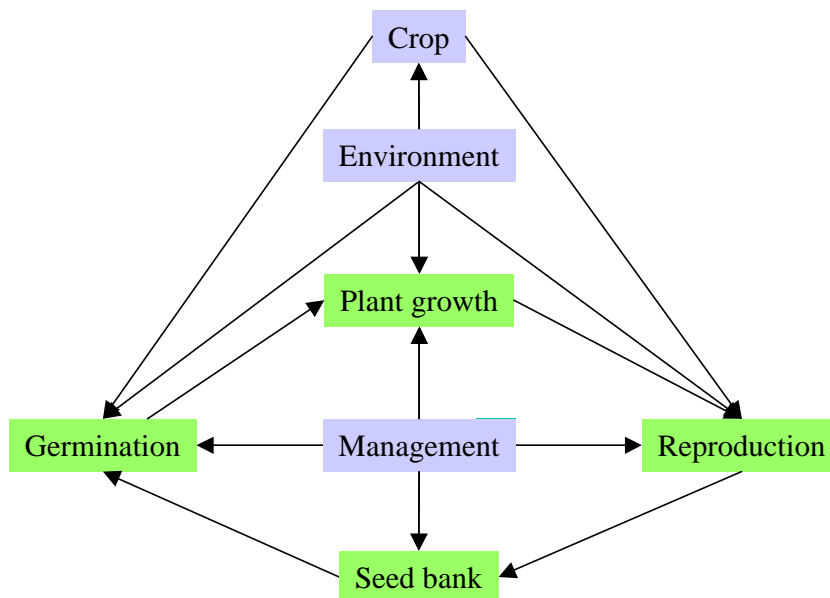
```
<model name="my model" based_upon="url(basics.xml)/model">
  <submodel type="weed" name="Poa annua">
    <submodel type="plant growth" method="Holst 2008">
      <parameter name="juvenile duration" average="554.5"/>
      Details of submodel...
    </submodel>
  </submodel>
  <submodel type="weed" name="Stellaria media">
    Details of submodel...
  </submodel>
</model>
```

(1) For this parameter the “average” value in the base model (a WTDB lookup) is overridden with the value 554.5.

(2) The ‘Stellaria media’ submodel did not exist in the base model and will therefore be added.

(3) This model did not redefine the ‘Viola arvensis’ submodel of the base model. As a result, the final model will contain a ‘Viola arvensis’ submodel identical to that of the base model.

3. Submodels



A is influenced by B

The figure shows the WeedML submodels and how they typically interact in model logic. Submodels shown in green lie nested inside the weed submodel (not shown).

WeedML defines a set of legal submodel types. The major ones are shown in the figure above and described in detail below.

For each submodel WeedML defines its intended behaviour, including its interaction with other submodels.

WeedML may define alternative behaviours of a submodel. The behaviour is chosen in the WeedML by the submodel's "method" attribute.

A collection of simple functions will be defined in WeedML ('linear', 'power', 'log-normal' ...) to aid model construction.

Future versions of WeedML may add alternative methods to existing submodels and also completely new types of submodels.

3.1 WEED

The main purpose of the weed submodel is to act as a container of the submodels, representing the four life phases of the weed.

3.1.1 SEED BANK

The seed bank submodel represent seeds in or upon the soil. Most simply this contains two parameters: the half-life (in years) of seeds in the soil and the daily mortality rate for seeds on the soil.

```
<submodel type="seed bank">  
  <parameter name="halflife in soil" avg="7.5" />  
  <parameter name="mortality on soil" avg="6.5%" />  
</submodel>
```

This submodel gets input from the reproduction submodel and provides input to the germination submodel. In addition the management submodel may cause a perturbation of seeds among soil layers.
– None of these interactions necessitates additional seed bank parameters.

3.1.2 GERMINATION

The germination submodel transfers seeds from the seed bank to the plant growth submodel. Seeds are transferred when they have started an irreversible process of germination.

Several methods of calculating germination have been published. Here, we define a very simple one, namely a germination calendar combined with sensitivity to soil depth, crop leaf area index (LAI), and maximum yearly germination rate at optimal conditions. This gives us the following parameters:

A monthly germination calendar:

- monthly relative germination (12 numbers)

Only the relative size of the numbers matters; they may add up to 1 or whatever. The daily potential germination is found by piece-wise linear interpolation between the 12 monthly points and scaling the whole curve to achieve a certain max. yearly germination rate:

- maximum yearly germination at optimal soil depth (percentage)

The relative germination by soil depth is described by a smooth curve. In literature, three different functions, which take either one or two parameters, have been used:

- name of function ('exponential', 'log-normal', 'Fermi-Dirac')
- 1 or 2 shape parameters

Germination is reduced by crop shadowing. We assume a negative exponential relation to crop LAI described by one parameter:

- exponential germination reduction by crop LAI

This could be described in WeedML like this:

```
<submodel type="germination" method="germination calendar"
  resolution="monthly">
  <parameter name="calendar" avg="(0 0 1 3 7 7 2 0 0 0 0)"/>
  <submodel name="germination by depth" type="log-normal">
    <parameter name="average" average="3.2"/>
    <parameter name="sd" average="0.452"/>
  </submodel >
  <submodel name="germination by LAI" type="exponential">
    <parameter name="exponent" average="-2.5"/>
  </submodel >
  <parameter name="optimal germination" average="30%"/>
</submodel>
```

3.1.3 PLANT GROWTH

The plant growth submodel keeps track of plant growth and development from when the seeds enter (from the germination submodel) and until plants reach the end of flowering. The reproduction submodel gets information from the plant growth submodel about the state of flowering to calculate the daily seed production rate.

The method applied in this submodel follows Holst (2008). It holds four life stages (germinating seed, seedling, juvenile, reproductive). Each life stage has two parameters to describe its duration plus some parameters to describe its physiological time scale:

- average duration
- variance of duration
- name of physiological time scale ('Days', 'Day-degrees', 'Photothermal', etc.)
- any number of parameters for the physiological time scale ('T0', 'Tmax', ...)

To transform emerging seedlings into 'seedling density equivalents' (i.e. matching the competitiveness of seedlings emerging at the same time as the crop), we use an empirical relation to crop LAI:

- 2 shape parameters for seedling competitiveness, $y = \exp[a(\exp(-x/b)-1)]$

For example, in WeedML:

```
<submodel type="plant growth" method="Holst 2008">
  <submodel name="DD0" type="time scale" method="day-degrees">
    <parameter name="T0" average="0"/>
  </submodel>
  <submodel name="DD4" type="time scale" method="day-degrees">
    <parameter name="T0" average="4"/>
  </submodel>
  <submodel name="PT8" type="time scale" method="photo-thermal">
    <parameter name="T0" average="8"/>
  </submodel>
  <submodel name="germinating seed">
    <parameter name="duration" average="213" variance="9.8"
      units=".../submodel[@name='DD0']"/>
  </submodel>
  <submodel name="seedling">
    <parameter name="duration" average="54" variance="2.8"
      units=".../submodel[@name='DD4']"/>
  </submodel>
  <submodel name="juvenile">
    <parameter name="duration" average="457" variance="60.2"
      units=".../submodel[@name='PT8']"/>
  </submodel>
  <submodel name="reproductive">
    <parameter name="duration" average="110" variance="23" units=".../DD4"/>
  </submodel>
  <submodel name="competition by LAI" type="double-exponential">
    <parameter name="a" average="1.8"/>
  </submodel>
```

```

    <parameter name="b" average="6.7"/>
  </submodel >
</submodel>

```

(1) Physiological time scales are defined as submodels.

(2) The duration of life stages are given in units defined by a 'time scale' submodel.

3.1.4 REPRODUCTION

The reproduction submodel gets information about the state of flowering from the plant growth submodel to calculate the daily seed production rate. The potential seed production is reduced due to competition from the crop and the weed community as a whole.

The method applied in this submodel follows Holst (2008). It holds 2 life stages describing seeds on the plant: immature and mature seeds. Their duration is described like the life stages of the plant growth submodel.

Seed production is calculated from the reproductive weed biomass:

- 2 allometric shape parameters for fecundity, $y=ax^b$.

For example, in WeedML:

```

<submodel type="reproduction" method="Holst 2008">
  <submodel name="DD4" type="time scale" method="day-degrees">
    <parameter name="T0" average="4"/>
  </submodel>
  <submodel name="immature seeds">
    <parameter name="duration" average="45" variance="7.5"
      units=".../submodel[@name='DD4']"/>
  </submodel>
  <submodel name="mature seeds">
    <parameter name="duration" average="77" variance="19.2"
      units=".../submodel[@name='DD4']"/>
  </submodel>
  <submodel name="fecundity" type="power">
    <parameter name="a" average="3.57"/>
    <parameter name="b" average="0.92"/>
  </submodel >
</submodel>

```

3.2 CROP

The crop submodel modifies the weed germination, plant growth and reproduction submodels through competition. These submodels work back by causing yield loss.

In the applied method, all these interactions are derived from the Cousens curves for yield loss (one curve for each crop×weed combination) and time-varying crop LAI (one phenology curve for each crop). Additional parameters are needed for the maximum (i.e. weed-free) yield and exchange rate of weed biomass to crop loss:

- Cousens's a and i (for each crop×weed)
- max. yield (kg/ha)
- exchange rate of weed biomass to crop loss

The exchange rate seems to be only little variable (0.6-0.8), so a crop-wise value should be sufficient.

```

<submodel name = "lupin" type="crop" method="yield loss/LAI">
  <submodel name="DD0" type="time scale" method="day-degrees">
    <parameter name="T0" average="0"/>
  </submodel>
  <submodel type="time line" name="LAI"> (1)
    <submodel type="time line" name="after sowing"> (2)
      <parameter name="start" value="sowing" units="management procedure"/>
      <parameter name="xy"
        value="( (0 0) (150 0) (300 0.2) (400 1) (500 2.5) (600 5) (870 5)
          (1400 5) (1850 2))"

```

```

        units=" .././submodel[@name=' DD0' ]" />
</submodel>
<submodel type="time line" name="after harvest">
  <parameter name="start" value="harvest" units="management procedure" />
  <parameter name="xy"
    value="(0 0.2)"
    units=" .././submodel[@name=' DD0' ]" />
</submodel>
<parameter name="exchange rate" average="0.7" />
<parameter name="a" average="70%" weed="Galium aparine" />
<parameter name="i" average="0.5%" weed="Galium aparine" />
<parameter name="a" average="15%" weed="Poa annua" />
<parameter name="i" average="0.04%" weed="Poa annua" />
</submodel>

```

(1) A a 'time line' submodel is simply a collection of 'time lines'; or,

(2) a 'time line' submodel defines a y (in this case crop LAI) as a function of x (in this case time in day-degrees). It contains a list of (x,y) values to be connected by interpolation. The beginning of a time line is either a management procedure or some other point in time, e.g. a calendar date.

The names of crops, weeds and management procedures must exist in the proper tables (3.5).

3.3 MANAGEMENT

The management submodel potentially affect all weed submodels: seed bank, germination, plant growth and reproduction.

The crop rotation is defined as a sequence of 'crop' submodels inside the 'management' submodel.

Each 'crop' submodel consists of a collection of 'procedure' submodels, each described by its kind, timing, the depth of soil tillage (if applicable) and the mortality caused on seedling, juvenile and adult weed plants (zero, if not given).

```

<submodel type="management" name="ww-ww-sb">
  <submodel name="Winter wheat 1" type="crop">
    <submodel type="procedure"> (1)
      <parameter name="kind" value="ploughing" depth="20" /> (2)
      <parameter name="time" units="date" average="17/09/01" format="D/M/Y"> (3)
    </submodel>
    <submodel type="procedure">
      <parameter name="kind" value="sowing" depth="4" />
      <parameter name="time" units="date" average="18/09/01" format="D/M/Y">
      <submodel name="mortality" method="3 stages"> (4)
        <parameter name="weeds" value="(apesv galap)" /> (5)
        <parameter name="seedling" average="90%" />
        <parameter name="juvenile" average="40%" />
        <parameter name="adult" average="0%" />
      </weed_mortality>
    </submodel>
    <submodel type="procedure">
      <parameter name="kind" value="herbicide-spraying" />
      <parameter name="time" units="date" average="09/10/01" format="D/M/Y">
      <submodel name="mortality" method="3 stages">
        <parameter name="weeds" value="*" /> (6)
        <parameter name="seedling" average="90%" />
        <parameter name="juvenile" average="40%" />
        <parameter name="adult" average="0%" />
      </weed_mortality>
    </submodel>
    <submodel type="procedure">
      <parameter name="kind" value="harvest" />
      <parameter name="time" units="date" average="01/08/02" format="D/M/Y">
      <submodel name="mortality" method="3 stages">
        <parameter name="weeds" value="*" />
        <parameter name="seedling" average="0%" />
        <parameter name="juvenile" average="0%" />
        <parameter name="adult" average="100%" />
      </weed_mortality>
    </submodel>
  </submodel>

```

```

    </submodel>
</submodel>

<submodel name="Winter wheat 2" type="crop" based_upon="winter wheat 1"/> (7)
<submodel name="Spring barley undersown" type="crop"
  based_upon="url(default_model)/model/submodel[@name='management']/
  submodel[@name='Spring barley undersown']"/>
</submodel>

```

- (1) Management procedures are defined as submodels. In this example they are unnamed (but not untyped).
- (2) The kind of procedure must exist in the management procedures table (3.5.3).
- (3) The time of application is defined – in this example on a calendar basis but a physiological time scale could also be used. Calendar years are relative to the current crop.
- (4) Weed mortality is defined as a submodel. In the example, a model specifying the mortality of three weed life stages is used.
- (5) The affected weeds are listed. The names must exist in the weed table (3.5.1).
- (6) An asterisc ('*') is used as a shorthand for all species.
- (7) The management of second year's winter wheat is defined as identical to the first year's.

3.4 ENVIRONMENT

The environment submodel supplies information about the weather and environment in general:

```

<submodel type="environment">
  <parameter name="longitude" value="11.845"/>
  <parameter name="latitude" value="65.901"/>
  <submodel name="Flakkebjerg, 30 years average" type="weather" (1)
    method="text file"/>
    <parameter name="filename" value="flakkebjerg.txt" has_header="yes"/>
  </submodel>
</submodel>

```

- (1) A submodel of type 'weather' encapsulates weather variables. In the first version of WeedML, the 'has_header' attribute must be set to 'yes'. It means that the weather file begins with a text section that describes the format of the weather file.

Other methods for the weather submodel include e.g. constant conditions:

```

<submodel name="hot" type="weather" method="constant"/>
  <parameter name="temperature" value="27" variance="2.5"/>
</submodel>

```

3.5 COMMON TABLES

Some common tables hold information that is usually shared among many models. The tables are part of the model definition. For example:

```

<model name="Reduced tillage" based_upon="url(default_model.xml)/model">
  <table name="weeds"
    based_upon="url(default_model.xml)/models/table[@name='weeds']"/>
  <table name="crops"
    based_upon="url(default_model.xml)/models/table[@name='crops']"/>
  <table name="management"
    based_upon="url(default_model.xml)/models/table[@name='management']"/>

  Rest of model definition...
</model>

```

3.5.1 Weed table

All weed names in the model description must be found in the weed table. Most conveniently you rely on a central table used by all models but you can also add additional entries in the table locally.

Here is an example of a shared table in file “global_labels.xml”:

```
<table name="weeds">
  <weed>
    <name text="Chenopodium album" language="lat"/>
    <name text="Lambsquarters" language="eng"/>
    <name text="Hvidmelet gåsefod" language="dan"/>
    <name text="CHEAL" language="bay"/>
  </weed>
  <weed>
    <name text="Stellaria media" language="lat"/>
    <name text="Chickweed" language="eng"/>
    <name text="Fuglegræs" language="dan"/>
    <name text="STEME" language="bay"/>
  </weed>

  More weeds...

</table>
```

Locally you can add new species or additional names to those in the base table. Here we add one new species (*Tussilago farfara*) and Swedish names for two existing species in the base table:

```
<table name="weeds" based_upon="url(global_labels.xml)/table">
  <weed>
    <name text="Tussilago farfara" language="lat"/>
    <name text="Coltsfoot" language="eng"/>
    <name text="Følfod" language="dan"/>
    <name text="TUSFA" language="bay"/>
    <name text="Hästhov" language="swe"/>
  </weed>
  <weed>
    <name text="STEME" language="bay"/>
    <name text="Våtarv" language="swe"/>
  </weed>
  <weed>
    <name text="CHEAL" language="bay"/>
    <name text="Svinmålla" language="swe"/>
  </weed>
</table>
```

The language is coded by abbreviations according to the ISO 639-3 standard (www.sil.org/iso639-3), with the addition of “bay” for Bayer codes (www.eppo.org/PUBLICATIONS/eppt/bayer_codes.htm).

3.5.2 Crop table

The crop table is defined along the same lines as the weed table:

```
<table name="crops">
  <crop>
    <name text="Winter wheat" language="eng"/>
    <name text="Vinterhvede" language="dan"/>
    <name text="TRZAW" language="bay"/>
  </crop >
  <crop>
    <name text="Carrots" language="eng"/>
    <name text="Gulerødder" language="dan"/>
    <name text="DAUCA" language="bay"/>
  </crop>

  More crops...

</table>
```

3.5.3 Management submodels table

The management procedures table is defined along the same lines as the weed table:

```
<table name="management">
  <procedure>
```

```
    <name text="Ploughing" language="eng"/>
    <name text="Pløjning" language="dan"/>
</procedure>
<procedure>
  <name text="Harvest" language="eng"/>
  <name text="Høst" language="dan"/>
</procedure>

More procedures...

</management>
```