Nomad - Module Definition DTD

by Christian Schneider

Table of contents

1. Module Definition DTD v1.0

1.1. Purpose

This DTD was developed to define the modules of the Nord Modular synthesizer and their hierarchie.

Note:

Any information represented in the XML file is independend of the language you chose for programming. The definitions of substitutions are not directly included into this DTD as the may contain language dependend information like class-names.

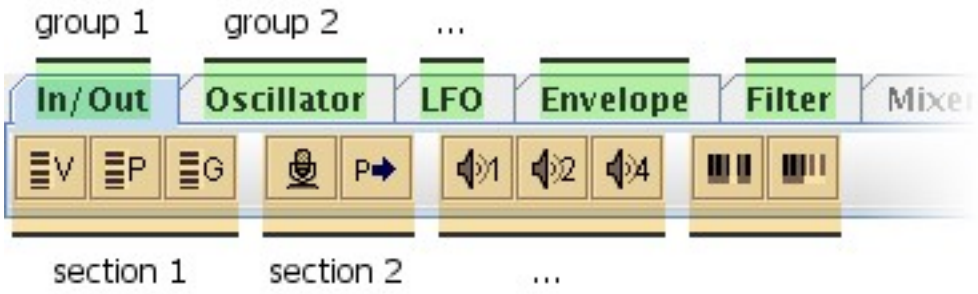
1.2. Structuring Modules

Along with defining the attributes of a single module this DTD supports adding information about how they relate to each other and how they are grouped.

Example: (contained elements: structure, group, section insert)

```
<structure>
  <group name="In/Out">
    <section>
      <insert module-id="1" />
      <insert module-id="63" />
      <insert module-id="65" />
    </section>
    <section>
      <insert module-id="2" />
      <insert module-id="127" />
    </section>
    ...
  </group>
  ...
</structure>
```

The provided structure can be used to generate the following tabbed pane.



[figure:tabbedpane - groups and sections]

- Each 'group'-element is mapped to a tab-group in the same order found in the xml file.
- The 'section'-element groups modules inside a tab. Two sections should be separated by a space.
- Modules inside a section should appear in the tab in the same order found in the xml file.

1.3. Proposal for new versions

The substitutions have to be better integrated. For example on an oscillator-module the frequency value (and with that it's substitution) depends on the states of frequency-display, coarse and fine. In this version of the DTD supports no substitutions that have more than one variable.

All possible names for a substitution should be defined in this DTD.

Language-independend substitutions for the values of a parameter should be defined in this DTD. (for example simple string-substitutions like: 0='Off',1='On')

Substitutions now threaded as language-depended, such that have to be calculated (for example the Oscillator's frequency) might be represented in the xml file as well. This could be realized as shown in the next example. (a similar substitution can be found in the substitutions DTD)

```
<substitution name="example-transformation">
    <transform
      offset        = "0"      <!-- default:0  , shortcut:o -->
      factor        = "1.0"    <!-- default:1.0, shortcut:f -->
      prefix-string  = "x"      <!-- default:"" , shortcut:p -->
      suffix-string  = "[UNIT]" <!-- default:"" , shortcut:s -->
    />
</substitution>
```

represents the transformation:

$$t_{o,f,p,s} := \begin{cases} int \longrightarrow string \\ value \longrightarrow concatenate(prefix, (value + offset) \cdot factor, suffix) \end{cases}$$

[figure:example transformation]

There could be an 'optional'-element containing optional information about a module. For example it could contain layout related definitions such as how elements should be grouped. For example

- coarse - fine - frequency-display belong together
- in- or outputs often belong to an additional level-knob

This would definately increase the development time required to generate custom themes.

Simply because

- grouped elements can automatically be positioned together
- more user interface components can be generated automatically for example grouped elements could be put into a groupbox. Also text/icon-labels (where reasonable).

1.4. Object Model

[figure:mapping]

1.5. modules-definition-v10.dtd - Reference

1.5.1. Top-level element(s)

- modules-definition

1.5.2. List of elements

- group
- input
- insert
- module
- modules
- modules-definition
- output
- parameter
- section
- structure
- true

1.5.3. Element declarations

group	
Content model:	(section +)
Used inside	structure
Attributes:	name — type: CDATA short-name — type: CDATA Optional attribute which must contain a shortcut for attribute 'nam

	'name' contains more than eight characters.
input	
Content model:	EMPTY
Used inside	module
Attributes:	id — type: NMTOKEN The id's value is equivalent to the index of an element defined by inputs with indices 0 and 1 must have an id with the same values name — type: CDATA type — possible values: audio, control, logic, slave
insert	
Content model:	EMPTY
Used inside	section
Attributes:	module-id — type: NMTOKEN The id of a module. A module with same id must be defined inside
module	
Content model:	(custom | input | output | parameter) *
Used inside	modules
Attributes:	id — type: NMTOKEN The id's value is equivalent to the index of an element defined by inputs with indices 0 and 1 must have an id with the same values short-name — type: CDATA cva-only — possible values: true, false This attribute is only used in the PolyAreaIn module. Its value is fa

	only to be specified if it should have 'true' as value.
	limit: type: NMTOKEN
	Defines a limit of how many times a module is available in one pa means that the number of this module is not limited.
	cycles: type: NMTOKEN
	dyn-mem: type: NMTOKEN
	height: type: NMTOKEN
	prog-mem: type: NMTOKEN
	x-mem: type: NMTOKEN
	y-mem: type: NMTOKEN
	zero-page: type: NMTOKEN

modules	
Content model:	(module +)
Used inside	modules-definition
Attributes:	- none -
modules-definition (top-level element)	
Content model:	(structure, modules)
Used inside	- none -
Attributes:	- none -
output	
Content model:	EMPTY
Used inside	module

Attributes:		
	id	type: NMTOKEN
	The id's value is equivalent to the index of an element defined by inputs with indices 0 and 1 must have an id with the same values	
	name	type: CDATA
	type	possible values: audio, control, logic, slave

parameter	
Content model:	EMPTY
Used inside	module

Attributes:		
	id	type: NMTOKEN
	The id's value is equivalent to the index of an element defined by inputs with indices 0 and 1 must have an id with the same values	
	name	type: CDATA
	min	type: NMTOKEN
	The smallest possible value. If the attribute is missing it implies a	
	max	type: NMTOKEN
	The largest possible value.	
	default	type: NMTOKEN
	The default value. This should be used to set the state of a param instantiated.	
	bit-count	type: NMTOKEN
	Number of bits reserved for the value. (expected by the protocol)	
	use-substitution	type: CDATA
	Contains an identifier for the substitution that should be used to g the parameter's raw value. A missing 'use-substitution' attribute in to it's substitution. This simply means that for example the numbe	

	(Find out more about substitutions.)
section	
Content model:	(insert +)
Used inside	group
Attributes:	- none -
structure	
Content model:	(group +)
Used inside	modules-definition
Attributes:	- none -
true	
Substitutions are not supported in this DTD version for element 'custom'.	
Content model:	EMPTY
Used inside	- none -
Attributes:	id — type: NMTOKEN min — type: NMTOKEN The smallest possible value. If the attribute is missing it implies a max — type: NMTOKEN The largest possible value. default — type: NMTOKEN The default value. This should be used to set the state of a param instantiated. bit-count — type: NMTOKEN Number of bits reserved for the value. (expected by the protocol)

2. Substitution DTD v1.0

2.1. Purpose

The idea behind substitutions is to XMLize the transformation from a parameter's numeric representation to its string-representation. This has several convenient advantages:

- less information must be included in source code
- automated linking of substitutions and parameters

2.2. Proposal for new versions

See the "Proposal for new versions"-section.

2.3. substitutions-v10.dtd - Reference

2.3.1. Top-level element(s)

- substitutions

2.3.2. List of elements

- if
- item
- list
- substitutions
- transform
- use-implementation

2.3.3. Element declarations

if	
If a parameter's value is equal to attribute 'value-is', 'replace-with' supersedes the superior substitution.	
Content model:	EMPTY
Used inside	transform
Attributes:	replace-width — type: CDATA value-is — type: NMTOKEN

item	
Content model:	(#PCDATA)
Used inside	list
Attributes:	- none -

list		
Contains a list of strings. The parameters value is used as index to an 'item' element where the 0 is the index for the first item, 1 is the index for the second item, ...		
Content model:	item +	
Used inside	substitutions	
Attributes:	id	type: ID
	Name of the substitution	

substitutions (top-level element)		
Content model:	(transform + , use-implementation + , list +)	
Used inside	- none -	
Attributes:	id	type: ID
	Name of the substitution	

transform		
This element defines the transformation t(value) := concatenate(prefix, (value+offset)·factor, suffix) . If element 'if' matches value, it supercedes the transformation t(value).		
Content model:	if ?	
Used inside	substitutions	
Attributes:	id	type: ID
	Name of the substitution	
	factor	type: NMTOKEN

	offset	type: NMTOKEN
	prefix	type: CDATA
	suffix	type: CDATA

use-implementation		
This element links a custom implementation with a parameter.		
Content model:	EMPTY	
Used inside	substitutions	
Attributes:	id	type: ID
	Name of the substitution	
	class	type: ID
	Identifier for the custom implementation.	