the RIF slides Format (source) It is intended to be a 3D enhanced version of Microsoft Powerpoint. We admit however that our socalled .pvl format lacks many of Powerpoint original features. But, on the other hand, Powerpoint does not offer many of the features we provide.

the pvl format language We use a simplified form of **HTML**, with some additional features.

example

```
<h4>A header</h4>
<ul>

item 1 </li>

item 2 </li>
</ul>
```

A description of the origin of this approach can be found at

http://*www.cs.vu.nl/~eliens/online/hush/manuals/man/man8/ht-ml.html*
http://*www.cs.vu.nl/~eliens/online/papers/htf4*

displaying slides The slides can be displayed in a number of ways:

- in latex, text slides
- in (dynamic) HTML, for presentation
- in VRML, for presentation and the annotation of virtual worlds

To realize presentations a collection filters is needed, among which the *vr-ml.flt* For how the slides are used see also:

http://*www.cs.vu.nl/~eliens/online/guide.html*

presentation level In VRML display mode, there may be multiple levels in a slide. Levels are used to allow for incremental presentation of the content of slide. Each element in a slide, may have an indication of *start* and *end* level. Some tags allow for a indicating automatic increments of the *start* level, which is usually indicated with *level=auto*.

creating slides There are a number of contexts, and associated file types, in which slides may be created:

- .t – the ultimately generic format, uses all filters
- .tm – limited .t format with VRML support, uses ht-ml.flt + vr-ml.flt
- .pvl – VRML slides support, uses vr-ml.flt -pvl
- .es – for the use of slides in worlds

See example makefiles for actual usage

parameters for vr-ml.flt

- none – plain generation of VRML
- -pvl – generates pvl includes
- -x – creates also links for hrefs (UDF)

afterthoughts on formatting The VRML protos supporting the slides format must be kept simple and basic. In some cases extra tags are needed to get the desired result (or an approximation of that).

extra tags

- <+*n*> – create n additional horizontal space (float n)
- ←*n*> – for negative spacing

beware of

- always use a <p> tag after ending a list or display.

Also, the filtering is done on an as best as possible basis. In some cases it will be very hard to get what you want. Actually, since it is a legacy format, you're not even supposed to use it. So, why bother.

on the design of the tag structure To accomodate the various modes of formatting and the various uses that can be made of the slide format, we make a number of assumptions:

- processing (in presentation mode) should be fast
- the tags must be intuitive, and easy to type (keyboard-wise)
- the tags must support a sufficiently rich rethoric repertoire

the slide tag

- id – unique name (obligatory)

- caption – slides caption (for use in latex)
- mode – UDF (undocumented feature)
- style – generic style attribute (UDF)
- transform attributes – trans, scale, rotate
- (VRML) style tags – bgcolor, textcolor

the text tag (VRML only)

- transform attributes – trans, scale, rotate
- (VRML) style tags – bgcolor, textcolor

¡slideid=sl-uol¿

the list tags: ul and ol Both unnumbered and numbered list are allowed, as in HTML

- level – with level=auto the bullets will appear in order
- additional attributes – UDFs

To indicate items in a list use the <*li*> tag. When additional lines are need, use the *line* tag.

beware (¡red¿may/must change (MC)¡/red¿) The text after a bullet or number must be on one line, additional lines need to be indicated with a tag.

the *dl* list tag Should have

- level

Use the <dt> and <dd> tags to create the items for the list.

displays To allow for unformatted text, such as program text the *display* tag is supported, which is like the tag in HTML.

- level – with level=auto lines will appear in order
- size – to give the size of the text used in the display

the font tag

- color
- size
- style

- family

font abbreviations

- <i>...</i> – *italic*
- <b>...</b> – **bold**
- <em>...</em> – *emphasize*
- <tt>...</tt> – `typewriter`
- <rm>...</rm> – roman

font color abbreviations

- <red>...</red> – red
- <blue>...</blue> – blue
- <yellow>...</yellow> – yellow
- <green>...</green> – green
- <white>...</white> – white
- <black>...</black> – black
- <gold>...</gold> – gold
- <silver>...</silver> – silver

the *object* tag

¡break size=1¿ It might be worthwhile to incorporate VRML objects in a slide. These are only visible in VRML, naturally. As an example:

```
<object¿
Transform { translation 9 0 -4 children [
<include file=sensor.wrl>
] }
</object¿
```

Note that the VRML text is automatically hidden when using the object tag. The *include* tag is a feature of the .t format.

hiding VRML The <*vrml*> and </*vrml*> tags are needed if you don't want your VRML code to be displayed in HTML.