herm-pic - Graphical extension package for LaTeX2e to produce HER-Diagramms

Sven Schoradt
(sven@dex.de)

September 15, 2005

1 General remarks

The `herm-pic` package, helps you to draw Higher Entity Relationship Diagramms. This diagramms are the visualisation of HER schemas, a database modeling concept. For more information to this see the original paper at http://citeseer.nj.nec.com/thalheim91foundation.html.

2 Package information

You use the `herm-pic` Package in Version 1.0.0(1.5) from the 2005/09/15 .

This information and some more about the package state you can request from the revision macros defined in `herm-rev.sty`. This are

HPdescription prints a short description of the package

HPrevision prints the full revision information, consisting major, minor and releaseversion and a releasecomment like `'-rc2'`

HPlongrevision prints also the version information and the cvs revision number of the `herm-rev.sty` file for consistency checks

HPdate prints the date the revision is last changed

3 Usage

To use the package you only have to insert the line

```
\usepackage{herm-pic}
```

into the header part of your document. There are no known problems with other packages or classes. If you find something in this manner, mail to the package author.

3.1 Package options

The package accepts the options `verbatim`, `center`, `box` and `erd`.

verbatim prints some processing information to the log file, its mainly usable for debugging purposes

center center the scema verticaly on the page

box draw a box with the exact dimensions of the schema around the schema, so you can see if some elements will be drawn outside the schema boarders

erd draws Entity-Relationship Diagramms instead of HERM Diagramms, see for more information the HermPic ER Diagramm documentation

4 The schema environment

In the `herm-pic` package I defined the `schema` environment.

This environment is the standard way to use the schema elements, but you can also use the schema elements in an ordinary picture environment.

You use the schema environment in the following manner:

```
\begin{schema}(x,y)

\end{schema}
```

where `x` is the width and `y` is the height of the schema.

To scale the schema you can use the `\hermpic` length. You set it with

```
\setlength{\hermunit}{.6cm}
```

5 The schema elements

To most of the schema element macros there exists a star version that provides an interface with to point placing and with unlimited names. For the nonstar version you can give a valid internal name as an optional parameter.

The names of object, that are used for internal things should not consist any active tex stuff, that means no macros, no escaped chars (like the undersgore) and the on. To garantee that there are no limits to the printed object names you can give optional an internal name without these things.

5.1 Entities

To use an entity in your schema, you can use the `\entity` macro. The syntax is the following:

```
\entity(x,y){text}
```

,

```
\entity[internal_name](x,y){text}
```

or

```
\entity*(x,y){text}
```

where `(x,y)` are the coordinates of the left lower edge of the entity symbol. The `text` was printed as the entity name. In the first version the entity name should only consits of Characters and Numbers, but no tex special chars are allowed. In the other version there is no limitation to the text.

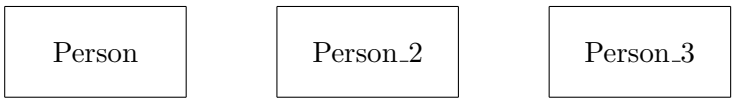
The internal name should only consits of characters and numbers.

The entity symbol will be a rectangle with the width of 4 and the height of 2 units.

Example:

```
\begin{schema}(17,4)
\entity(1,1){Person}
\entity[person2](7,1){Person\_2}
\entity*(12,1){Person\_3}
\end{schema}
```

produces this



5.2 Relationships

To draw relationships in your diagrams, you can use the `\relation` macro. The syntax is the following:

```
\relation(x,y){text}
```

,

```
\relation[internal_name](x,y){text}
```

or

```
\relation*(x,y){text}
```

where `(x,y)` are the coordinates of the left edge. The `text` will be displayed as the name of the relationship. There are the same restrictions to the name how they are descriped for entitys.

Example:

```
\begin{schema}(17,4)
\relation(1,1){Relation}
\relation[relation2](7,1){Relation\_2}
\relation*(12,1){Relation\_3}
\end{schema}
```

produces this

5.3 Cluster

To use the cluster concept in your schema, you can use the `\cluster` macro. The syntax is as followed:

```
\cluster(x,y){name}
```

,

```
\cluster[internal_name](x,y){name}
```

or

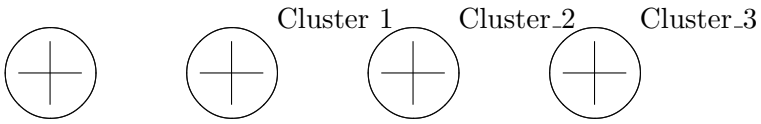
```
\cluster*(x,y){name}
```

This draws the cluster item, thats a circle with the radius 2 whith a plus in it, on the position `(x,y)` and places the name right over the cluster.

Example:

```
\begin{schema}(16,4)
\cluster(1,1){}
\cluster(5,1){Cluster 1}
\cluster[cluster2](9,1){Cluster\_2}
\cluster*(13,1){Cluster\_3}
\end{schema}
```

produces this

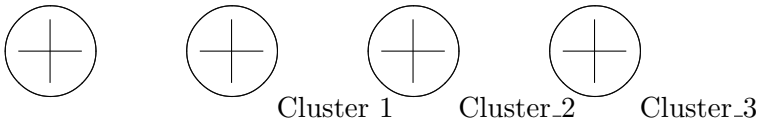


To slect the position where the cluster name is displayed, you can use the `\cnamepos` macro. The possible values are the same like at the attribute possitioning (see there).

Example:

```
\begin{schema}(16,4)
\cnamepos{ru}
\cluster(1,1){}
\cluster(5,1){Cluster 1}
\cluster[cluster2](9,1){Cluster\_2}
\cluster*(13,1){Cluster\_3}
\end{schema}
```

produces this



5.4 Attributes

Attributes in HERM will be a string connected with the entity or relation. To draw attributes with herm-pic you have to use the macro `\attr` with the syntax:

```
\attr[pos]{objectname}{attributename}
```

or

```
\attr*[pos](x,y){attributename}
```

In the non-star version you have only 12 docking points to the object, setable with the pos attribute. Figure 1 shows the positionings.

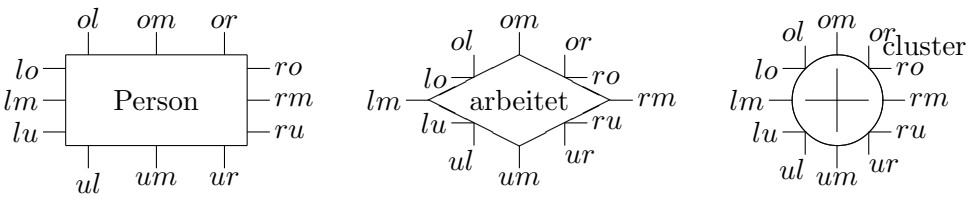


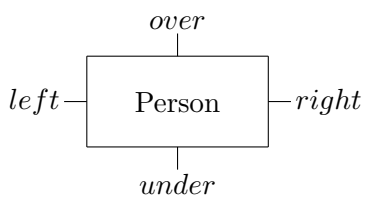
Figure 1: Positioning posibilities for attributes

in the star version you can set the docking position with the (x,y) coordinates and the alignment to the object with the pos attribute (l,r,o,u).

Example:

```
\begin{schema}(10,6)
\entity(4,2){Person}
\attr[lm]{Person}{left}
\attr[om]{Person}{over}
\attr[rm]{Person}{right}
\attr[um]{Person}{under}
\end{schema}
```

produces



To show that an attribute or an sub-attribute is a key concept for the object, entity or relationship, you can use the `\key` macro, like this:

```
\begin{schema}(10,4)
\entity(1,1){Person}
\attr*(4,3)[r]{\key{key}}
\end{schema}
```

produces this

6 Connections between schema elements

A connection between two schema elements is a vector with an optional text marking for the cardinality constraint.

The package provides for this the `\connection` macro, with the syntax

```
\connection(objectname1,objectname2){mathematical equation}
```

or

```
\connection*(x1,y1)(x2,y2){mathematical equation}
```

It draws a vector from one object to an other. The star version draws a vector from the point (x1,y1) to (x2,y2) and puts the mathematical equation to an adequate point at the end of the vector.

For the lazy guys there is a short name for theis macro, named `\conn`.

Example:

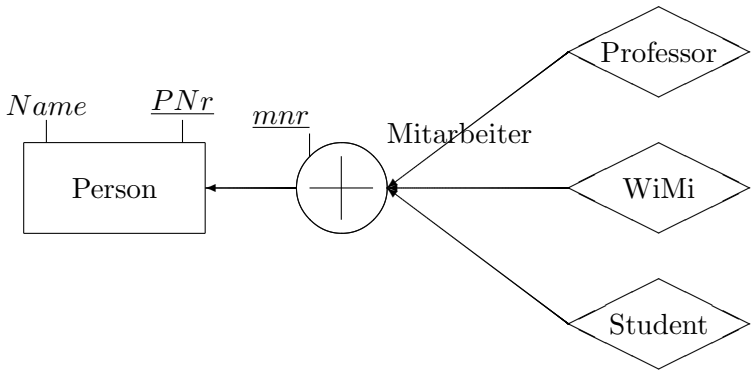
```
\begin{schema}(20,5)
  \entity(1,1){Person}
  \entity(13,1){Team}
  \relation[arbeitetin](7,1){arbeitet\_in}
  \connection(arbeitetin,Person){(0,3)}
  \connection(arbeitetin,Team1){(2,10)}
\end{schema}
```

produces

,

```
\begin{schema}(15,10)
  \entity(1,4){Person}
  \attr[ol]{Person}{Name}
  \attr[or]{Person}{\key{PNr}}
  \relation(13,7){Professor}
  \relation(13,4){WiMi}
  \relation(13,1){Student}
  \cluster(7,4){Mitarbeiter}
  \attr[ol]{Mitarbeiter}{\key{mnr}}
  \connection(Mitarbeiter,Person){}
  \connection(Professor,Mitarbeiter){}
  \connection(WiMi,Mitarbeiter){}
  \connection(Student,Mitarbeiter){}
\end{schema}
```

produces



and

```
\begin{schema}(20,5)
  \entity(1,1){Person}
  \entity(13,1){Team}
  \relation*(7,1){arbeitet\_in}
  \connection*(7,2)(5,2){(0,3)}
  \connection*(11,2)(13,2){(2,10)}
\end{schema}
```

produces this

7 A more or less complex example

This more or less complex example shows you the usage of all the macros together.

```
\setlength{\hermunit}{.5cm}
\begin{schema}(30,15)
\entity(15,8){Dokument}
\attr[lo]{Dokument}{titel}
\attr[ro]{Dokument}{gueltig\_ab}
```

```
\attr[lu]{Dokument}{datei}
\attr[ru]{Dokument}{gueltig\_bis}
%
\entity(6,12){Satzung}
\attr[om]{Satzung}{\key{snummer}}
%
\entity(24,12){Kategorie}
\attr[rm]{Kategorie}{\key{kname}}
%
\entity(0,4){Amt}
\attr[rm]{Amt}{\key{aname}}
%
\entity(15,0){Wort}
\attr[ro]{Wort}{\key{wort}}
\attr[ru]{Wort}{soundex}
%
\relation(18,12){in}
\conn(in,Dokument){}
\conn(in,Kategorie){}
%
\relation(12,12){zu}
\conn(zu,Dokument){(1,1)}
\conn(zu,Satzung){}
%
\relation(24,8){unter}
\conn*(24,9)(24,12){}
\conn*(28,9)(28,12){}
%
\relation(0,0){uebergeordnet}
\conn*(0,1)(0,4){(0,1)}
\conn*(4,1)(4,4){}
%
\relation(0,12){verantwortlich}
\conn(verantwortlich,Satzung){(1,1)}
\conn(verantwortlich,Amt){}
%
\relation(12,4){Schlagwort}
\conn(Schlagwort,Dokument){}
\conn(Schlagwort,Wort){}
%
\relation(18,4){Inhalt}
\attr[rm]{Inhalt}{anzahl}
\conn(Inhalt,Dokument){}
\conn(Inhalt,Wort){}
```

```
\end{schema}
```

produces the schema