

herm-pic - Graphical extension package for LaTeX2e to produce HER-Diagramms The Entity-Relationship Diagramm mode

Sven Schoradt
(sven@dex.de)

September 15, 2005

1 General remarks

For a wider view on HermPic see the HermPic documentation.

2 The schema elements

All of the macros from the normal herm-pic you find also in the ERD mode. Some, especially `\cluster` are not known in ERD diagrams. In the ERD mode there are also some special macros that you not find outside this mode.

So herm-pic schemas are compatible to the ERD mode, but ERD diagrams work not in the normal herm-pic environment.

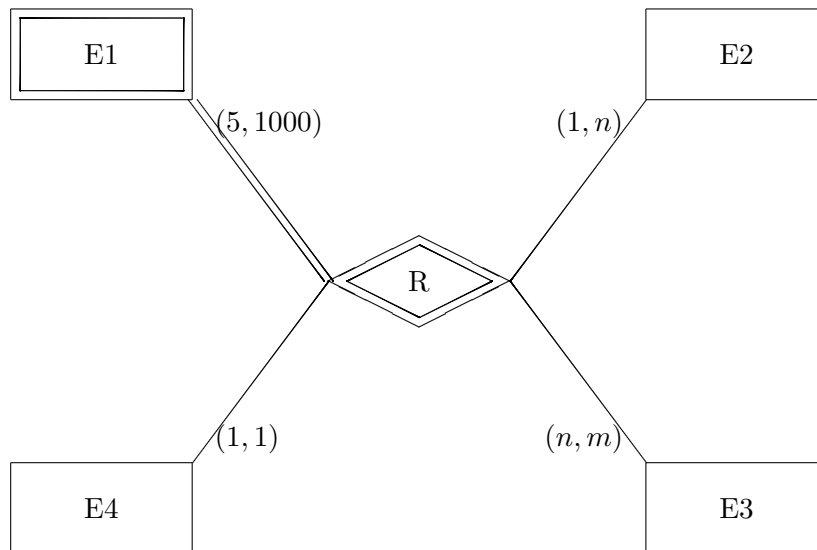
On entities and relations there are no changes, but there are some new object types.

That are weak entity and weak relation types, that work like the corresponding strong types with the macros `\wentity` and `\wrelation`.

```
\begin{schema}(20,14)
  \wentity(1,11){E1}
  \entity(15,11){E2}
  \entity(15,1){E3}
  \entity(1,1){E4}
  \wrelation(8,6){R}
  \conn(R,E1){(5,1000)}
  \conn(R,E2){(1,n)}
  \conn(R,E3){(n,m)}
```

```
\conn(R,E4){(1,1)}
\end{schema}
```

produces



And there is also the is-a type.

2.1 Is A

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

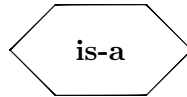
```
\isa(x,y){name}
```

This draws an is-a sign with the height of 2 and a width of 4 units. The name is only for internal use and have to be unique.

Example:

```
\begin{schema}(16,4)
\isa(6,1){isa1}
\end{schema}
```

produces this



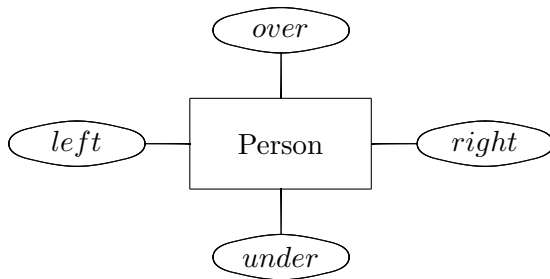
2.2 Attributes

Attribute are drawn as an oval with the name in it conected with the object.

Example:

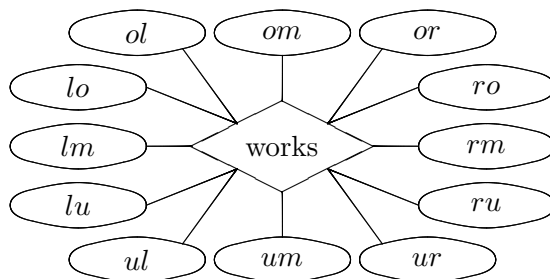
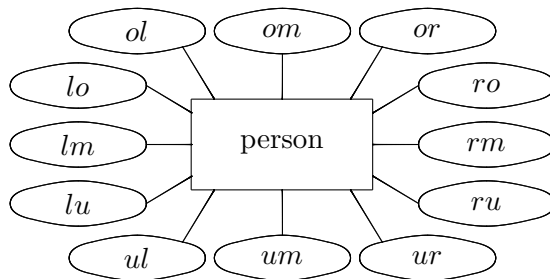
```
\begin{schema}(10,6)
\entity(4,2){Person}
\attr[lm]{Person}{left}
\attr[om]{Person}{over}
\attr[rm]{Person}{right}
\attr[um]{Person}{under}
\end{schema}
```

produces



and





2.3 Special attribute types

Special attribute types are the multivalued attribute (`\multivaluedattr`), the optional attribute (`\optattr`) and derived attributes (`\derivedattr`).

There are also structured attributes called `\complexattr`. The name of a complex attribute should not consist of special letters.

A structured attribute can consist of normal attributes, multivalued attributes and derived attributes.

They were used like simple attributes but there is no star or plus version available.

Example:

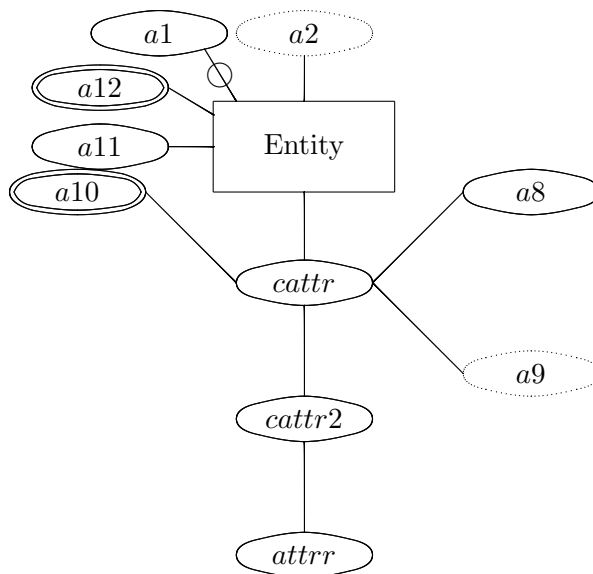
```
\begin{schema}(14,30)
  \entity[entity](5,23){Entity}
  \attr[lm]{entity}{a11}
```

```

\multivaluedattr[lo]{entity}{a12}
\optattr[ol]{entity}{a1}
\derivedattr[om]{entity}{a2}
\complexattr[um]{entity}{cattr}
\attr[ro]{cattr}{a8}
\derivedattr[ru]{cattr}{a9}
\multivaluedattr[lo]{cattr}{a10}
\complexattr[um]{cattr}{cattr2}
\attr[um]{cattr2}{attrr}
\end{schema}

```

produces



3 Connections between schema elements

Connections are drawn without arrows in ER diagrams. Only if you use a connection to or from an is-a type, there will be drawn arrows.

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



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



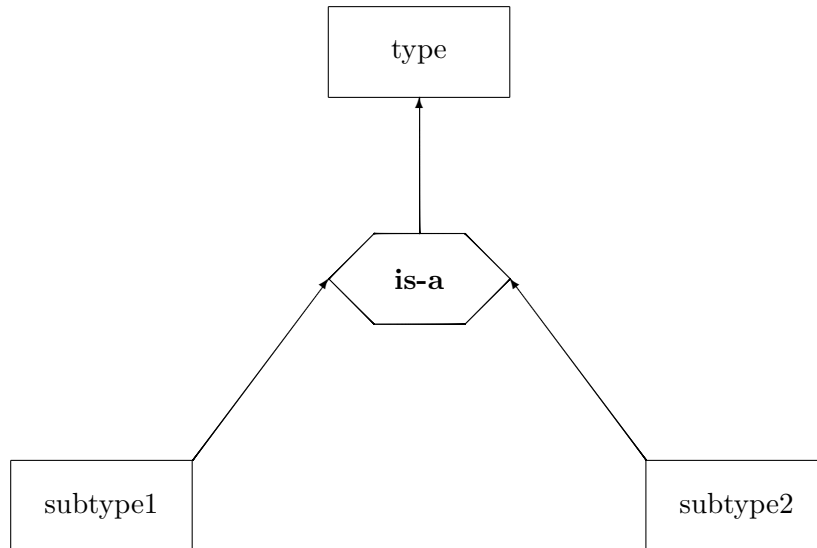
and

```

\begin{schema}(20,14)
\entity(1,11){Unter1}
\entity(15,11){Unter2}
\isa(8,6){isa1}
\entity(8,1){Ober}
\conn(Unter1,isa1){}
\conn(Unter2,isa1){}
\conn(isa1,Ober){}
\end{schema}

```

produces



4 Specialising

To implement ER specialising you can use the macros

```

\dspecial(x,y){iname}
\dspecial*(x,y){iname}
\ospecial(x,y){iname}
\ospecial*(x,y){iname}

\sconn(oname1,oname2)

```

The `\dspecial` macro represents an disjunct specialisation, the `\ospecial` macro an overlapping specialisation. The star versions represents the total versions of the specialisation.

The `\sconn` macro represents the specilisation between two objects. The objects must give in the order subtype, supertype.

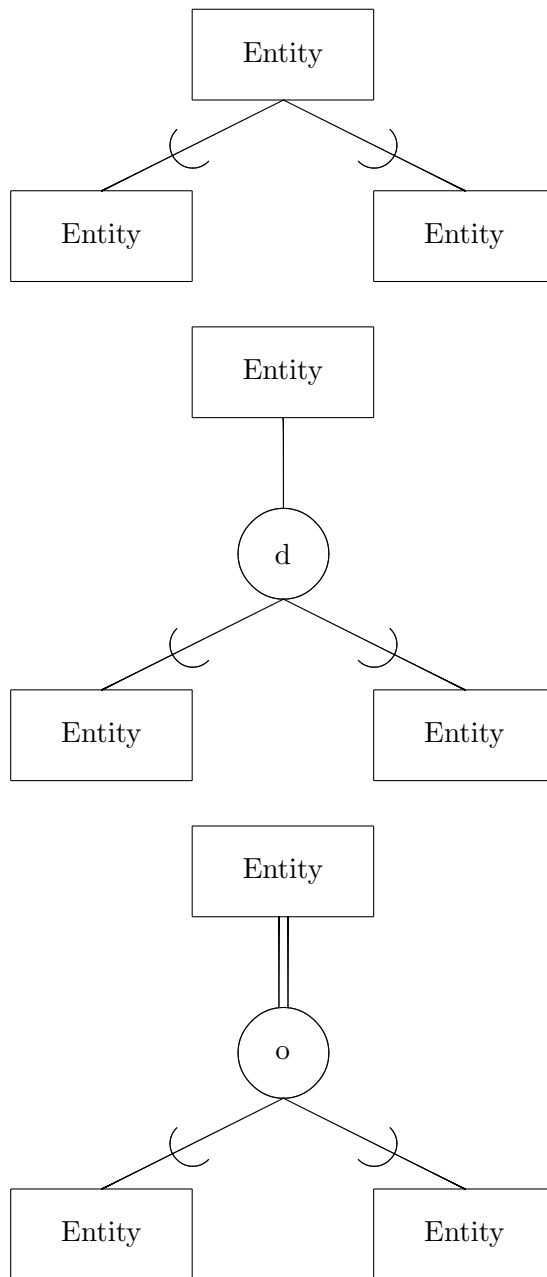
This works like this:

```

\begin{schema}(14,30)
  \entity[entity7](5,26){Entity}
  \entity[entity8](1,22){Entity}
  \entity[entity9](9,22){Entity}
  \sconn(entity8,entity7)

```

```
\sconn(entity9,entity7)
\entity[entity](5,27){Entity}
\dspecial(6,23){test1}
\connection(entity,test1){}
\entity[entity2](1,19){Entity}
\entity[entity3](9,19){Entity}
\sconn(entity2,test1)
\sconn(entity3,test1)
\entity[entity4](5,16){Entity}
\ospecial*(6,12){test2}
\connection(entity4,test2){}
\entity[entity5](1,8){Entity}
\entity[entity6](9,8){Entity}
\sconn(entity5,test2)
\sconn(entity6,test2)
\end{schema}
```

5 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}(30,17)
\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

