The smartdiagram package*

Claudio Fiandrino
claudio.fiandrino@gmail.com

December 23, 2016

Abstract

This package provides a way to easily draw diagrams in documents and presentations from a list of items thanks to Ti*k*Z. The idea comes out from this question on TeX.StackExchange.

Contents

1 Introduction and requirements **2**

2 Basic Usage **3**

3 Additions **4**

4 The options **7**
4.1 Setting the options 7
4.2 Available options 7
4.2.1 General options 7
4.2.2 Specific options 8
4.2.3 Options of the additions library 11

5 Gallery of examples **12**

6 Recommendations and known issues **21**
6.1 Something about colors 21
6.2 Defining styles 22
6.3 Circular, bubble and constellation diagrams 22
6.4 Descriptive diagrams 23
6.5 Decorations 23
6.6 Priority descriptive diagrams 24

*This document corresponds to smartdiagram v0.3b, dated 2016/12/23; it is released under and subject to the LaTeX Project Public License (LPPL).

7 Aknowledgements **25**

8 Implementation **25**
8.1 Initialization and Package Options 25
8.2 Keys and color declaration 26
8.3 Commands 35
8.4 Library Additions 44

1 Introduction and requirements

The aim of the package is to provide a way to draw diagrams starting from a list of items colored automatically. The diagrams created could be used in a simple document or in a presentation: in the latter case, while using the Beamer class, the user could decide if diagrams should be overlay-awared or not.

Automatically, the smartdiagram package loads:

- Ti*k*Z;
- etoolbox;
- xparse;
- xstring (from version 0.2).

Moreover, the package loads the following Ti*k*Z libraries:

- `backgrounds`;
- `calc`;
- `fadings`;
- `shadows`;
- `shapes.arrows`;
- `shapes.symbols` (from version 0.2).

and it sets a new layer called `smart diagram arrow back`. From version 0.3, the package is composed of three core libraries:

- `core.definitions`,
- `core.styles`,
- `core.commands`

which actually form the package `smartdiagram.sty` and of the external library:

- `additions`

which can be loaded separately by the user. By loading this library:

```
\usesmartdiagramlibrary{additions}
```

forces the Ti*k*Z library `positioning` to be loaded as well.

The package could be loaded by means of `\usepackage{smartdiagram}`.

2 Basic Usage

`\smartdiagram` The basic command is `\smartdiagram[`⟨*type of diagram*⟩`]{`⟨*list of items*⟩`}`. The `{`⟨*list of items*⟩`}` should be comma-delimited: to insert, for example, a comma as part of the item label use `\smartdiagram[..]{elem1,{elem2,text},elem3}`.

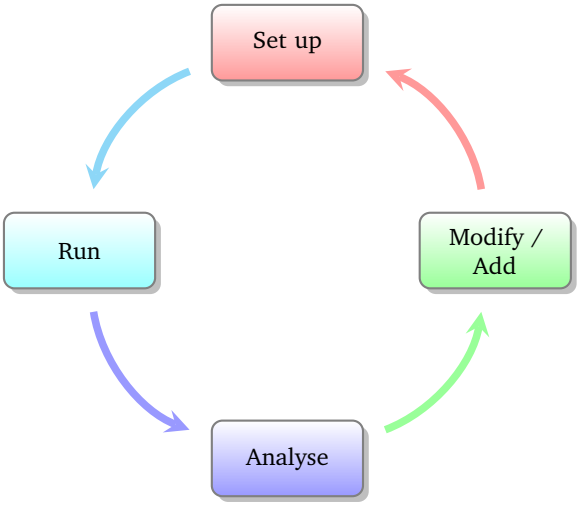
The possible diagrams that could be created are:

- `circular diagram`: the items in the list are diplayed around a circle typically in counterclockwise order;
- `circular diagram:clockwise`: same as before, but now items are placed in clockwise order (no space between names and `:`);
- `flow diagram`: the items in the list are diplayed as a flow chart;
- `flow diagram:horizontal`: the items in the list are diplayed as an horizontal flow chart (no space between names and `:`);
- `descriptive diagram`: a diagram in which are displayed concepts and their description;
- `priority descriptive diagram`: a diagram in which the items are deployted based on their relevance;
- `bubble diagram`: each item is a bubble deployted around a bubble center, which is the first element in the list;
- `constellation diagram`: each item is a circle connected to the center, the first element in the list again;
- `connected constellation diagram`: each item is a circle and, a part from the first element in the list, the other ones are connected toghether;
- `sequence diagram`: the items in the list are diplayed in sequence and each item points to the subsequent.

For example:

```
\begin{center}
\smartdiagram[circular diagram]{Set up,Run,Analyse,Modify~/ Add}
\end{center}
```

produces:



All diagrams could be customized in various ways, from the selection of the background colors to the font size, from the size of the items to the shape of the border. The keys necessary for this task will be analysed in detail in section 4.

For what concern the `bubble diagram`, the `constellation diagram` and the `connected constellation diagram`, they are a bit different from the rest of the diagrams, in the sense that the first item in the {⟨*list of items*⟩} is particular: called *bubble center* and *planet*, respectively, its aspect is different and it could be customized with dedicated keys.

From version 0.2 there is availble also the `sequence diagram` which displays the {⟨*list of items*⟩} with a particular shape.

`\smartdiagramanimated`

Inside presentations, the user could select if the diagram should be displayed in a *persistent* way, or with an animation. The *persistent* way is achieved by exploiting, again, `\smartdiagram`, while to have the diagram *overlay-aware* the command is `\smartdiagramanimated[`⟨*type of diagram*⟩`]{`⟨*list of items*⟩`}`.

For diagrams like the `circular diagram` and the `flow diagram` the animation runs as follows: at the beginning the first item of the list is displayed, then each time appears an arrow connecting the previous element with the new one; at the end it is displayed the arrow connecting the last element with the first one. Considering instead the `descriptive diagram`, per couple description title-description, first it is shown the description title and subsequently the description. In the `priority descriptive diagram`, very simply, the list of items is deployted starting from the bottom, that is the less relevant item, to the top to progressively show much relevant items. For what concern the `bubble diagram`, the `constellation diagram` and the `connected constellation diagram`, at first it is always shown the *bubble center* and *planet*, respectively and later all the items subsequently. Finally, in the `sequence` diagram, the items are presented one at a time starting from the left one ending with the right one.

3 Additions

`\usesmartdiagramlibrary`

From version 0.3, it is possible to load a separate library called `additions` which allows to create annotations over a smart diagram. Load the library through:

```
\usesmartdiagramlibrary{additions}
```

\smartdiagramadd in the preamble. The basic command introduced by the library is `\smartdiagramadd[`⟨*type of diagram*⟩`]{`⟨*list of items*⟩`}{`⟨*list of additions*⟩`}`. The `{`⟨*list of additions*⟩`}` have a special syntax:

⟨*position of module/Annotation text*⟩

where:

- *position* is an anchor of TikZ (i.e `above`, `below right` and so on);
- *module* is the name of a module in the smart diagram;
- *position* and *module* should be separated by the string `of`: spaces before and after the string are *mandatory*.

Smartdiagram defines as names:

- for the diagrams `circular diagram` and `circular diagram:clockwise`: `module`progressive-number (no space or other symbols in between); example: `module1`;
- for the diagrams `flow diagram` and `flow diagram:horizontal`: `module`progressive-number; example: `module3`;
- for the diagram `descriptive diagram`: `module-title`progressive-number for titles and `module`progressive-number for descriptions; example: `module-title1` and `module1`;
- for the diagram `bubble diagram`: `center bubble` for the center module and `module`progressive-number for the other modules; example: `center bubble` and `module2`;
- for the diagrams `constallation diagram` and `connected constellation diagram`: `planet` for the center module and `satellite`progressive-number for the other modules; example: `planet` and `satellite3`;
- for the diagram `priority descriptive diagram`: `module`progressive-number;
- for the diagram `sequence diagram`: `sequence-item`progressive-number; example: `sequence-item1`.

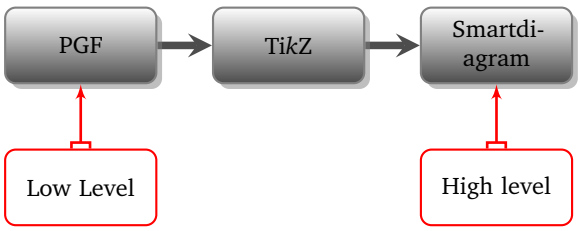
The use of the library `additions` in a document requires two compilation runs at least because of the TikZ options `remember picture` and `overlay`.

An example:

```
\begin{minipage}[t][3.5cm]{\textwidth}
\begin{center}
\smartdiagramset{
 uniform color list=gray!60!black for 3 items,
 back arrow disabled=true,
 additions={
  additional item offset=0.85cm,
  additional item border color=red,
  additional connections disabled=false,
  additional arrow color=red,
  additional arrow tip=stealth,
  additional arrow line width=1pt,
  additional arrow style=]-latex',
  }
```

```
}
\smartdiagramadd[flow diagram:horizontal]{%
PGF,Ti\textit{k}Z,Smartdiagram%
}{%
below of module1/Low Level, below of module3/High level%
}
\end{center}
\end{minipage}
```

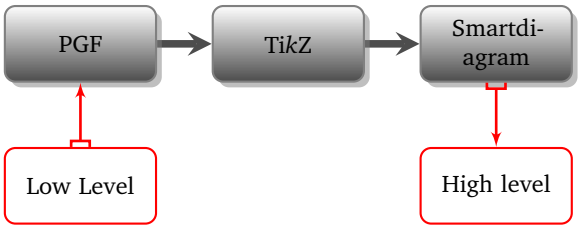
The result:



`\smartdiagramconnect` Notice that with the `\smartdiagramadd` facility it is not possible to fine customize the direction of the arrow tips. Since each additional module has as name `additional-module`*progressive-number*, then by means of the specific command `\smartdiagramconnect{`⟨*arrow options*⟩`}{`⟨*start module/end module*⟩`}` one could do better. For example:

```
\begin{minipage}[t][3.5cm]{\textwidth}
\begin{center}
\smartdiagramset{
 uniform color list=gray!60!black for 3 items,
 back arrow disabled=true,
 additions={
  additional item offset=0.85cm,
  additional item border color=red,
  additional arrow color=red,
  additional arrow tip=stealth,
  additional arrow line width=1pt,
  additional arrow style=]-latex',
  }
}
\smartdiagramadd[flow diagram:horizontal]{%
PGF,Ti\textit{k}Z,Smartdiagram%
}{%
below of module1/Low Level, below of module3/High level%
}
\smartdiagramconnect{{]-latex'}}{additional-module1/module1}
\smartdiagramconnect{{latex'-[}}{additional-module2/module3}
\end{center}
\end{minipage}
```

gives:



Because of the option `overlay`, it is better to protect within a `minipage` the diagram: in this way the missing bounding box update would not affect the subsequent/precedent text.

By default, the arrows are customized by the keys which start with `additional` (explained more in detail in the subsubsection 4.2.3), but it is possible to override locally this definition, for example with:

```
\smartdiagramconnect{{latex'-[},green}{additional-module2/module3}
```

For the moment, the library `additions` *does not work* in the animated mode.

4 The options

4.1 Setting the options

`\smartdiagramset` The options should be introduced similarly to what happens with `\tikzset` in TikZ: `\smartdiagramset{`⟨*list of options*⟩`}`. As well as in TikZ, it is possible to collect options within styles: see for more details subsection 6.2. Examples in which the options are used are shown in section 5.

4.2 Available options

Here follows the list of general available options: these options are related to generic aspects as color lists or arrows.

4.2.1 General options

- `set color list` (`initial: none`): this option allows the user to define the list of colors usable in the diagram;
- `uniform color list` (`initial: none`): this option allows the user to set one single color for the whole list of colors usable in the diagram;
- `use predefined color list` (`initial: none`): this option allows the user to reuse the pre-defined colors whenever in a previous diagram they have been changed;
- `insert decoration` (`initial: none`): this option allows to decorate the border of the module; the user should declare properly a decoration style and load manually the libraries required, e.g. `decorations.pathmorphing`; some more hints are given in section 6;

- `arrow line width` (`initial: 0.1cm`): this option sets the width of the connection arrows within two modules;
- `arrow tip` (`initial: stealth`): this option allows to select the single arrow tip; possible choices are described in the pgfmanual and for particular types it is recommended to load the library `arrows`;
- `arrow style` (`initial: <-`): this option allows the user to define a new style for the arrow; as well as the key `arrow tip`, see the pgfmanual to see a list of possible arrow styles;
- `uniform arrow color` (`initial: false`): this option, set to true, activates the possibility to use one single arrow color for all the connections;
- `arrow color` (`initial: gray`): this option, when the key `uniform arrow color` is set to true, allows to select the uniform arrow color.

4.2.2 Specific options

Here follows the list of specific available options per type of diagram.

For what concern the `circular diagram`, `circular diagram:clockwise`, the `flow diagram` and the `flow diagram:horizontal`:

- `module minimum width` (`initial: 2cm`): this option sets the minimum width of the module;
- `module minimum height` (`initial: 1cm`): this option sets the minimum height of the module;
- `module y sep` (`initial: 1.65`): this option sets a vertical distance factor among the modules in a `flow diagram`;
- `module x sep` (`initial: 2.75`): this option sets an horizontal distance factor among the modules in a `flow diagram:horizontal`;
- `module shape` (`initial: rectangle,rounded corners`): this option should be used to change the shape of the module, but the user should load manually the proper Ti*k*Z library; for example, to user an `ellipse`, load `shapes.geometric` in the preamble;
- `text width` (`initial: 1.75cm`): this option sets the text width inside the module;
- `font` (`initial: \small`): this option sets the module font;
- `border color` (`initial: gray`): this option sets the border color of the module;
- `text color` (`initial: black`): this option sets the text color of the module;
- `circular distance` (`initial: 2.75cm`): this option sets the radius of cirle around which the modules in a `circular diagram`;

Considering just the `flow diagram` and the `flow diagram:horizontal`, there is a specific option to disable the back arrow going from the final module to the first one and to set its distance from the modules:

- `back arrow distance` (`initial: 0.5`): the option sets the distance (it hold for both types);
- `back arrow disabled` (`initial: false`): the option, set to `true` disables the back arrow.

Similarly, in the circular diagram and in the circular diagram:clockwise, there is a specific option to disable the back arrow going from the final module to the first one:

- circular final arrow disabled (initial: false): the option, set to true disables the final connection.

For what concern the descriptive diagram and the priority descriptive diagram:

- descriptive items y sep (initial: 1.75): this option sets a vertical distance factor among the descriptive items;
- description title width (initial: 1.5cm): this option sets the minimum width of the description-title;
- description title text width (initial: 1.25cm): this option sets the text width of the description-title;
- description text width (initial: 5cm): this option sets the text width of the description;
- description title font (initial: \small): this option sets the font size of the description-title;
- description font (initial: \small): this option sets the font size of the description;
- description width (initial: 5.5cm): this option sets the minimum width of the description;
- priority arrow width (initial: 1.5cm): this option sets the width of the vertical arrow;
- priority arrow head extend (initial: 0.15cm): this option sets the width of the arrow head extend;
- priority tick size (initial: 5pt): this option sets the size of the line denoting the relevance position of items in the vertical arrow;
- priority arrow height advance (initial: 2cm): this option specifies how much the vertical arrow is vertically extended above the most relevant item.

For what concern the bubble diagram, the constellation diagram and the connected constellation diagram:

- bubble center node size (initial: 4cm): the option specifies the minimum size of the bubble center node;
- bubble center node font (initial: \large): this option sets the font size of the bubble center node;
- bubble center node color (initial: lightgray!60): this option allows to customize the background color of the bubble center node;
- distance center/other bubbles (initial: 0.8cm): this options specifies which is the distance among the bubble center node and the other bubbles; keep this value under controll to avoid that bubbles do not overlap anymore the bubble center node;
- distance text center bubble (initial: 0.5cm): this option sets the distance from the text to the border of the bubble center node;
- bubble fill opacity (initial: 0.5): this option sets the opacity at which the bubbles are shown;

- `bubble node size` (`initial: 2.5cm`): the option specifies the minimum size of the bubbles;
- `bubble text opacity` (`initial: 0.8`): this option sets the opacity at which the bubble text is shown;
- `bubble node font` (`initial: \normalfont`): this option sets the font size of the bubbles;
- `planet size` (`initial: 2.5cm`): the option specifies the minimum size of the planet;
- `planet color` (`initial: lightgray!60`): this option allows to customize the background color of the planet;
- `planet font` (`initial: \large`): this option sets the font size of the planet;
- `distance planet-connection` (`initial: 0.1cm`): this option sets the distance from the planet to the arrow directed to the satellites;
- `distance planet-text` (`initial: 0.5cm`): this option sets the distance from the text to the border of the planet;
- `planet text width` (`initial: 1.75cm`): this option sets the planet text width;
- `satellite size` (`initial: 1.75cm`): the option specifies the minimum size of the satellites;
- `satellite font` (`initial: \normalfont`): this option sets the font size of the satellites;
- `satellite fill opacity` (`initial: 0.5`): this option sets the opacity at which the satellites are shown;
- `satellite text opacity` (`initial: 0.8`): this option sets the opacity at which the satellite text is shown;
- `satellite text width` (`initial: 1.5cm`): this option sets the satellite text width;
- `distance satellite-connection` (`initial: 0.075cm`): this option sets the distance from the satellites to the arrows directed to the planet;
- `connection line width` (`initial: 0.1cm`): this option allows to customize the width of the connections from the planet to the satellites;
- `distance planet-satellite` (`initial: 3.5cm`): this option determine the distance among any pair of planet-satellite.

Considering just the the `constellation diagram` and the `connected constellation diagram`:

- `uniform connection color` (`initial: false`): the option, set to `true` overrides the color list definition;
- `connection color` (`initial: gray`): this option allows to specify the color valid for all the connections.

For what concern the `sequence diagram`:

- `sequence item height` (`initial: 1cm`): the option specifies the minimum height of the items;
- `sequence item width` (`initial: 2cm`): the option specifies the minimum width of the items;

- `sequence item border color` (`initial: gray`): sets the border line color;
- `sequence item border size` (`initial: 1.65\pgflinewidth`): sets the border line width;
- `sequence item font size` (`initial: \normalfont`): this option sets the font size of the items;
- `sequence item fill opacity` (`initial: 1`): sets the opacity at which the item is shown;
- `sequence item text opacity` (`initial: 1`): sets the opacity at which the item text is shown;
- `sequence item text width` (`initial: 1.9cm`): the option allows to select the item text width;
- `sequence item text color` (`initial: black`): the option allows to select the item text color;
- `uniform sequence color` (`initial: false`): the option, set to `true` overrides the color list definition;
- `sequence item uniform color` (`initial: gray!60!black`): this option allows to specify the color valid for all the items in the sequence.

4.2.3 Options of the additions library

The options of the library necessitate to be set within a specific key `additions`; this key is defined as

```
\pgfkeys{/smart diagram/.cd,
 additions/.style={/smart diagram/additions/.cd,#1}%
}
```

and it basically sets the correct path; indeed all of these keys are defined in a subtree of the main path: `/smart diagram/additions`. For example:

```
\smartdiagramset{
 additions={
  additional item offset=0.85cm,
  additional item border color=red,
  additional arrow color=red,
  additional arrow tip=stealth,
  additional arrow line width=1pt,
  additional arrow style=]-latex',
  }
}
```

Notice that each key starts with `additional`: in my opinion, although it may seems heavy type this every time, it could avoid some confusion with other keys.

- `additional item shape` (`initial: rectangle,rounded corners`): this option should be used to change the shape of the additional module; similarly to the key `module shape`, for some shapes the user should load manually the proper Ti*k*Z library;

- additional item border color (initial: none): sets the border line color;
- additional item bottom color (initial: white): this option sets the bottom color of the module; use it if you want to keep the same aspect of the diagrams: circular diagram, flow diagram, circular diagram:clockwise and flow diagram:horizontal;
- additional item fill color (initial: none): this option sets the fill color of the module; use it if you want to keep the same aspect of the other types of diagrams;
- additional item text width (initial: 1.75cm): it defines the text width of the module;
- additional item width (initial: 2cm): it defines the minimum width of the module;
- additional item height (initial: 1cm): it defines the minimum height of the module;
- additional item font (initial: \normalfont): this option allows to customize the font of the module;
- additional item border decoration (initial: none): this option allows to customize the module with a decoration; some more hints are given in section 6;
- additional item offset (initial: 0.25cm): this option defines the distance between the original module and the additional one;
- additional item fill opacity (initial: 1): it sets the fill opacity of the module;
- additional item text opacity (initial: 1): it sets the text opacity of the module;
- additional arrow tip (initial: stealth): this option defines the single arrow tip of the connection;
- additional arrow line width (initial: 0.1cm): this option defines the line width of the connection;
- additional arrow color (initial: gray): this option defines the connection color;
- additional arrow style (initial: ->): this option allows to customize the connection aspect, that is both arrow tips simultaneously;
- additional item shadow (initial: none): it allows to define a shadow for the module; use the usual TikZ shadow options;
- additional connections disabled (initial: true): this option when set to false makes all the connections visible.

5 Gallery of examples

Horizontal flow chart: custom color list - no back arrow

Example of an horizontal flow chart with custom color list and back arrow disabled:

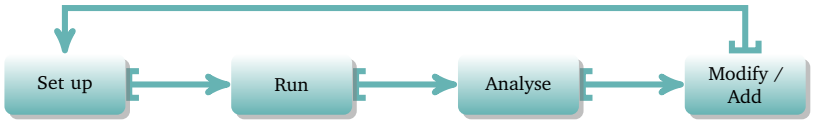
```
\begin{center}
\smartdiagramset{border color=none,
  set color list={blue!50!cyan,green!60!lime,orange!50!red,red!80!black},
  back arrow disabled=true}
```

```
\smartdiagram[flow diagram:horizontal]{Set up,Run,Analyse,Modify~/ Add}
\end{center}
```

`Horizontal flow chart: uniform color list - custom arrow style`

A similar example with an uniform color list and custom arrow style definition:

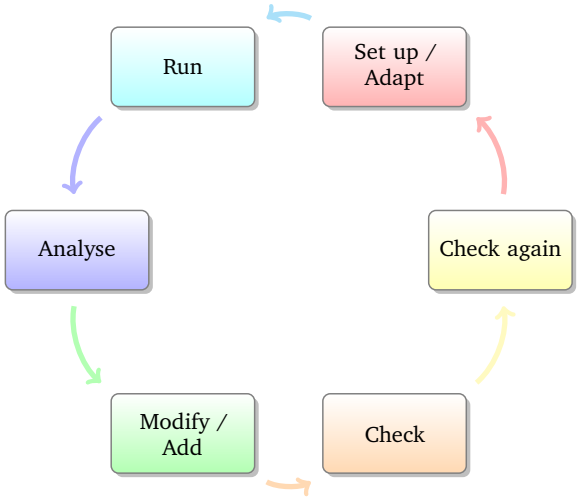
```
\begin{center}
\smartdiagramset{border color=none,
  uniform color list=teal!60 for 4 items,
  arrow style=[-stealth',
  module x sep=3.75,
  back arrow distance=0.75,
}
\smartdiagram[flow diagram:horizontal]{Set up,Run,Analyse,Modify~/ Add}
\end{center}
```



`Circular diagram with custom options`

Another example:

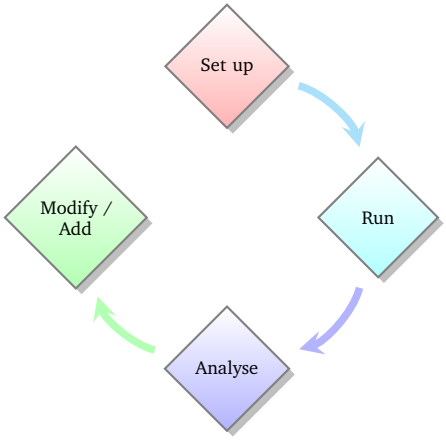
```
\begin{center}
\smartdiagramset{circular distance=4cm,
  font=\large,
  text width=2.5cm,
  module minimum width=2.5cm,
  module minimum height=1.5cm,
  arrow tip=to}
\smartdiagram[circular diagram]{Set up~/ Adapt,Run,Analyse,Modify~/ Add,
 Check,Check again}
\end{center}
```



`Circular diagram with custom shape and final arrow disabled`

An example with a diamond shape and the final arrow disabled:

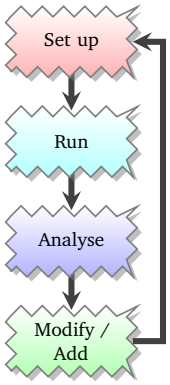
```
\begin{center}
\usetikzlibrary{shapes.geometric} % required in the preamble
\smartdiagramset{module shape=diamond,
 font=\scriptsize,
 module minimum width=1cm,
 module minimum height=1cm,
 text width=1cm,
 circular distance=2cm,
 circular final arrow disabled=true,
}
\smartdiagram[circular diagram:clockwise]{Set up,Run,Analyse,Modify~/ Add}
\end{center}
```



`Flow diagram with decorated border and uniform arrow color`

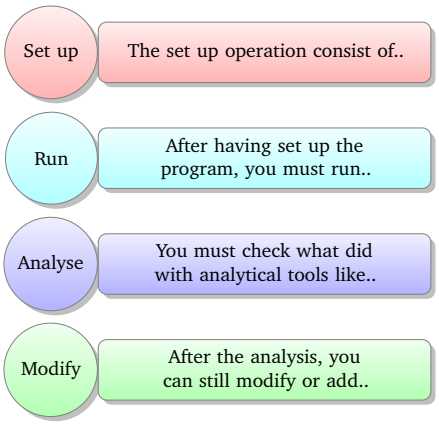
An example with a decorated shape and uniform arrow color:

```
\usetikzlibrary{decorations.pathmorphing} % required in the preamble
\begin{center}
\tikzset{my decoration/.style={decorate,decoration=zigzag}}
\smartdiagramset{module shape=rectangle,
  insert decoration={my decoration},
  uniform arrow color=true,
  arrow color=gray!50!black,
}
\smartdiagram[flow diagram]{Set up,Run,Analyse,Modify~/ Add}
\end{center}
```



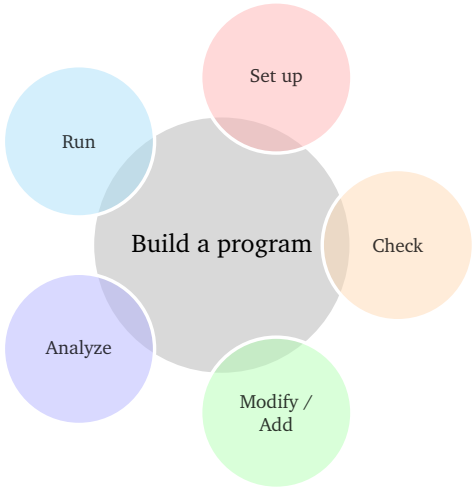
`Descriptive diagram` An example of descriptive diagram:

```
\begin{center}
\smartdiagram[descriptive diagram]{
{Set up,The set up operation consist of..},
{Run, {After having set up the program, you must run..}},
{Analyse, You must check what did with analytical tools like..},
{Modify, {After the analysis, you can still modify or add..}},
}
\end{center}
```



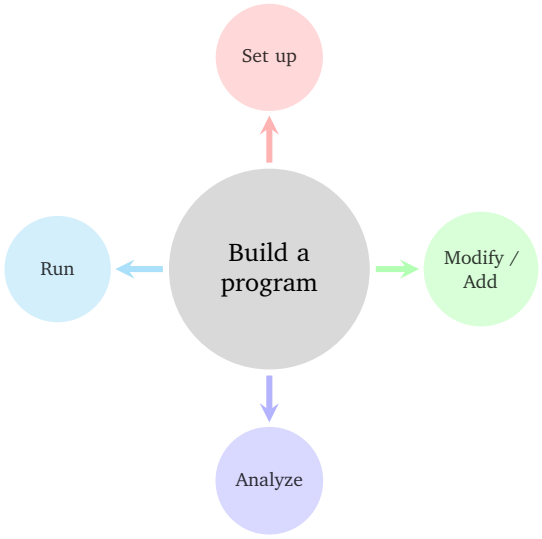
`Bubble diagram` An example of bubble diagram:

```
\begin{center}
\smartdiagram[bubble diagram]{
 Build a program,Set up,Run,Analyze,Modify~/\\ Add,Check
}
\end{center}
```



`Constellation diagram` An example of constellation diagram:

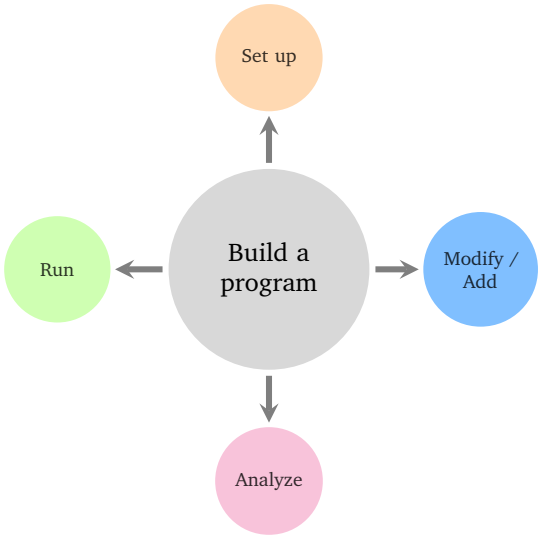
```
\begin{center}
\smartdiagram[constellation diagram]{
 Build a program,Set up,Run,Analyze,Modify~/\\ Add
}
\end{center}
```



`Constellation diagram with custom colors`

An example of constellation diagram with custom colors:

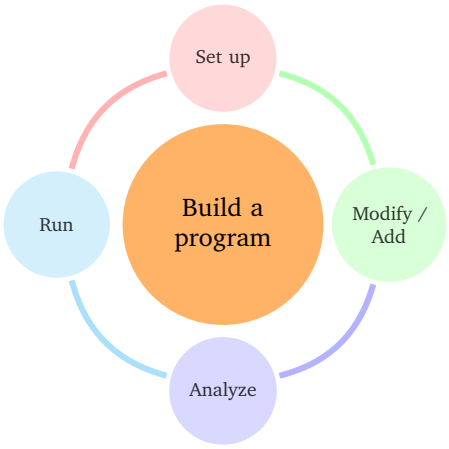
```
\begin{center}
\smartdiagramset{set color list={orange!60, green!50!lime!60,magenta!60,
  blue!50!cyan},
  uniform connection color=true
}
\smartdiagram[constellation diagram]{
 Build a program,Set up,Run,Analyze,Modify~/\\ Add
}
\end{center}
```



`Connected constellation diagram`

An example of connected constellation diagram:

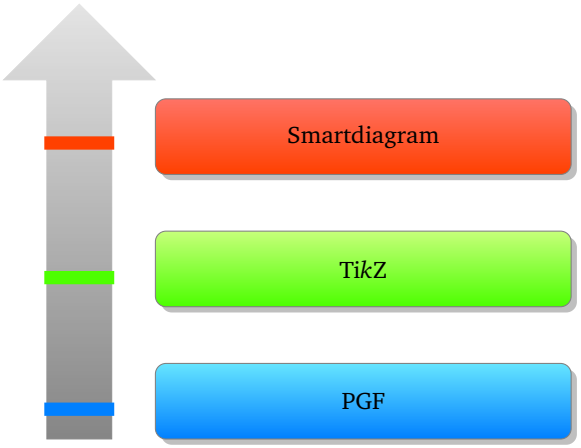
```
\begin{center}
\smartdiagramset{planet color=orange!60,
 distance planet-satellite=1cm
}
\smartdiagram[connected constellation diagram]
 {Build a program,Set up,Run,Analyze,Modify~/\\ Add,Check}
\end{center}
```



`Priority descriptive diagram`

An example of priority descriptive diagram describing that Ti*k*Z is built on top of PGF and Smartdiagram on top of Ti*k*Z:

```
\begin{center}
\smartdiagramset{
 set color list={blue!50!cyan,green!60!lime,orange!50!red},
 priority arrow width=2cm,
 priority arrow height advance=2.25cm
}
\smartdiagram[priority descriptive diagram]{PGF,Ti\textit{k}Z,Smartdiagram}
\end{center}
```



`Sequence diagram` The same previous example with a sequence diagram:

```
\begin{center}
\smartdiagram[sequence diagram]{PGF,Ti\textit{k}Z,Smartdiagram}
\end{center}
```

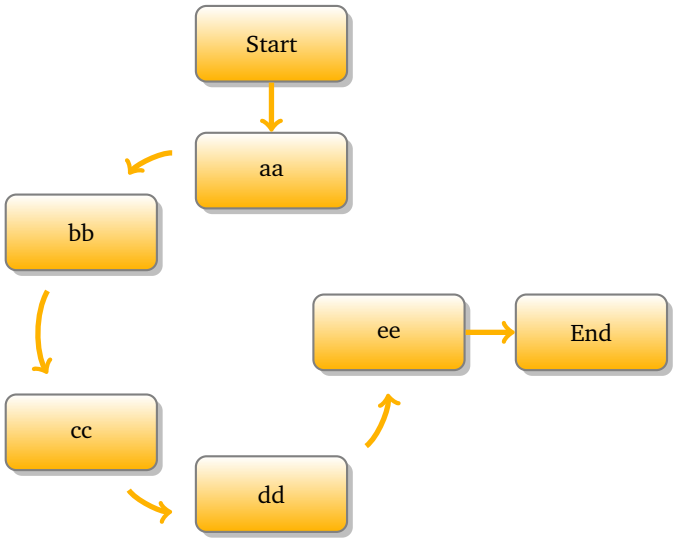
and with `uniform sequence color` set to `true`:

```
\begin{center}
\smartdiagramset{uniform sequence color=true,
  sequence item border color=black,sequence item font size=\footnotesize,
  sequence item text color=white
}
\smartdiagram[sequence diagram]{PGF,Ti\textit{k}Z,Smartdiagram}
\end{center}
```

`Put additions to a diagram` Here is an example of a circular diagram with some additions:

```
\usesmartdiagramlibrary{additions} % required in the preamble
\usetikzlibrary{arrows} % required in the preamble
\bigskip
\begin{minipage}[c][8cm]{\textwidth}
\centering
\smartdiagramset{
 uniform color list=orange!60!yellow for 5 items,
 circular final arrow disabled=true,
```

```
 circular distance=2.25cm,
 arrow tip=to,
 arrow line width=2pt,
 additions={
   additional item bottom color=orange!60!yellow,
   additional item border color=gray,
   additional item shadow=drop shadow,
   additional item offset=0.65cm,
   additional arrow line width=2pt,
   additional arrow tip=to,
   additional arrow color=orange!60!yellow,
 }
}
\smartdiagramadd[circular diagram]{
 aa,bb,cc,dd,ee
}{
 above of module1/Start,right of module5/End
}
\smartdiagramconnect{to-}{module1/additional-module1}
\smartdiagramconnect{-to}{module5/additional-module2}
\end{minipage}
```



6 Recommendations and known issues

6.1 Something about colors

As seen in the section 4, the colors could be customized by means of the key `set color list`. By default there are 10 predefined colors; in order: red!40, cyan!40, blue!40, green!40, orange!40, yellow!40, magenta!40, brown!40, violet!40 and teal!40. This implies that, by default, it is not possible to have lists longer than 10 items. This, of course, could be avoided by declaring proper lists with, say, 20 colors and therefore develop diagrams with more than 10 items. In any case, it is always possible to reset custom color lists by means of:

```
\smartdiagramset{use predefined color list}
```

In order to use one color for all the items it is possible to exploit the `uniform color list`; it has a particular syntax:

```
\smartdiagramset{uniform color list=<some color> for <n> items}
```

The ⟨*some color*⟩ is set for a list of ⟨*n*⟩ items and nothing more, so in order to avoid problems make sure you dimension ⟨*n*⟩ correctly. Indeed, in case ⟨*n*⟩ is lower than the number of items inside the diagram the following happens:

```
\begin{center}
\smartdiagramset{
 uniform color list=gray!60!black for 2 items,
 back arrow disabled=true,
}
\smartdiagram[flow diagram:horizontal]{PGF,Ti\textit{k}Z,Smartdiagram}
\end{center}
```

In conclusion, the uniform setting is extend only for ⟨*n*⟩ items, for the remaining ones the predefined or a custom color list is used.

Notice also that the key `uniform color list` makes the arrow color be uniform for ⟨*n*⟩ items, but it has no relation with the `uniform arrow color` which automatically makes *all* items with an uniform color. Indeed, the color taken by the arrows with:

```
\smartdiagramset{uniform arrow color=true}
```

could be customized throught the key `arrow color`, while `uniform color list` make modules and arrows be rendered with the same color.

6.2 Defining styles

The smartdiagram package admits the definition of styles to collect key-definitions; for example:

```
\smartdiagramset{my diagram style/.style={
       module shape=diamond,
       font=\scriptsize,
       module minimum width=1cm,
       module minimum height=1cm,
       text width=1cm
  }
}
```

can be subsequently used in:

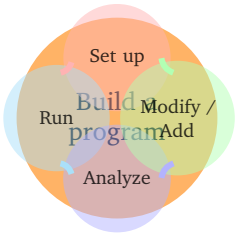
```
\begin{center}
 \smartdiagramset{my diagram style, arrow tip=to}
 \smartdiagram[circular diagram]{Do, This, Only, For, Me}
\end{center}
\begin{center}
 \smartdiagramset{my diagram style, module y sep=2.5}
 \smartdiagram[flow diagram]{Do, This, For, Me}
\end{center}
```

6.3 Circular, bubble and constellation diagrams

For these type of diagrams, the number of items is relevant: too many items lead to overlapping satellites and bubbles besides any attempt to resize things by means of keys that reduce the radius.

Notice also that imposing a too short distance from the planet to satellites is bad and leads to something like:

```
\begin{center}
\smartdiagramset{planet color=orange!60, distance planet-satellite=1cm}
\smartdiagram[connected constellation diagram]
 {Build a program,Set up,Run,Analyze,Modify~/\\ Add,Check}
\end{center}
```



6.4 Descriptive diagrams

When the user has to build a `descriptive diagram`, simple or animated, the following rules have to be respected:

- each description title and description should be separated by a comma;
- to use a comma inside a description, enclose by { } the description;
- use a comma after the last couple description title-description.

A working example:

```
\smartdiagram[descriptive diagram]{
{Set up, The set up operation consist of..},
{Run, {After having set up the program, you must run..}},
{Analyze, You must check what did with analytical tools like..},
}
```

A non-working example:

```
\smartdiagram[descriptive diagram]{
{Set up, The set up operation consist of..},
{Run, {After having set up the program, you must run..}},
{Analyze, You must check what did with analytical tools like..}
}
```

The *mandatory* final comma is missing thus, as result, the last couple description title-description will be entirely treated as a description title.

Another suggestion regarding descriptive diagrams is about the description title: it should be short in order to avoid the size of the circle explode. To kept it under controll, the keys `description title width`, `description title text width` and `description title font` are of help.

6.5 Decorations

To decorate the border of modules, it is kindly recommended to first declare the decoration choosen by means of an apposite style and then apply the style. The procedure, in code, should be as follows:

```
\tikzset{my wonderful decoration/.style={decorate,decoration=bent}}
\smartdiagramset{insert decoration=my wonderful decoration,...}
```

where the dots represent the other options.

The insertion of decorations inside a `descriptive diagram` are problematic: only `random steps`, `bent` and `coil` do not raise errors. Other decorations, like `snake`, raise as error:

```
! Dimension too large.
```

but, after all, the decoration is deployed anyway.

In case the user wants to decorate a border with a decoration that involves random numbers, such as `random steps` or other custom-built decorations, it is preferable to set a seed for the animated diagram, to avoid that at each step the border of the same module changes. An example:

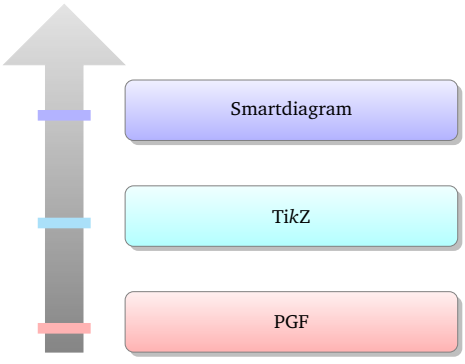
```
\begin{frame}
\begin{center}
\pgfmathsetseed{12354}
\tikzset{my decoration/.style={decorate,decoration=random steps}}
\smartdiagramset{insert decoration=my decoration}
\smartdiagramanimated[descriptive diagram]{
{Set up, The set up operation consist of..},
{Run, {After having set up the program, you must run..}},
{Analyze, You must check what did with analytical tools like..},
{Modify, {After the analysis, you can still modify or add..}},
}
\end{center}
\end{frame}
```

6.6 Priority descriptive diagrams

The vertical arrow is drawn in background with a fading effect: it may occurr that under some pdf editor this effect is not shown (this happened to me with evince, but it perfectly worked with okular).

For what concern these diagrams there is a bug in displaying the tick line when the `priority arrow head extend` is set with a bigger size than the default; a minimal-non-working example:

```
\begin{center}
\smartdiagramset{priority arrow width=2cm,
 priority arrow height advance=2.25cm,
 priority arrow head extend=0.3cm}
\smartdiagram[priority descriptive diagram]{PGF,Ti\textit{k}Z,Smartdiagram}
\end{center}
```



7 Aknowledgements

I would like to acknowledge first of all Alain Matthes and Mohsen because the `bubble diagram` and the `constellation diagram` are based on Alain's answer while the `circular diagram` is based on Mohsen's answer.

I would also like to thank Enrico Gregorio and Ahmed Musa for the courtesy of explaining me why my poor attempt in creating the `set color list` failed and for providing me valid solutions. Enrico also kindly fixed a spacing bug concerning the `uniform color list`.

Last, but not least, I would like to thank prof. Ludger Humbert for suggesting and providing the code for the `circular diagram:clockwise` as well as for pointing out some bugs in the version 0.3 and André Hilbig for suggesting the key `back arrow disabled`.

8 Implementation

8.1 Initialization and Package Options

This subsection highlights which are the package loaded and the `tikzlibraries` needed.

```
\RequirePackage{tikz}

\RequirePackage{etoolbox}
\RequirePackage{xparse}
\RequirePackage{xstring}

\usetikzlibrary{backgrounds,
   calc,
   fadings,
   shadows,
   shapes.arrows,
   shapes.symbols
}
\pgfdeclarelayer{smart diagram arrow back}
\pgfsetlayers{background,smart diagram arrow back,main}


\def\usesmartdiagramlibrary{\pgfutil@ifnextchar[{\use@smartdiagramlibrary}{\use@@smartdiagraml
\def\use@smartdiagramlibrary[#1]{\use@@smartdiagramlibrary{#1}}
\def\use@@smartdiagramlibrary#1{%
 \edef\pgf@list{#1}%
 \pgfutil@for\pgf@temp:=\pgf@list\do{%
  \expandafter\pgfkeys@spdef\expandafter\pgf@temp\expandafter{\pgf@temp}%
  \ifx\pgf@temp\pgfutil@empty
  \else
    \expandafter\ifx\csname smartdiagram@library@\pgf@temp @loaded\endcsname\relax%
    \expandafter\global\expandafter\let\csname smartdiagram@library@\pgf@temp @loaded\endcsnam
    \expandafter\edef\csname smartdiagram@library@#1@atcode\endcsname{\the\catcode`\@}
    \expandafter\edef\csname smartdiagram@library@#1@barcode\endcsname{\the\catcode`\|}
    \catcode`\@=11
    \catcode`\|=12
    \pgfutil@InputIfFileExists{smartdiagramlibrary\pgf@temp.code.tex}{}{
```

```
      \PackageError{smartdiagram}{I did not find the smartdiagram library '\pgf@temp'.}{}
    }%
    \catcode`\@=\csname smartdiagram@library@#1@atcode\endcsname
    \catcode`\|=\csname smartdiagram@library@#1@barcode\endcsname
  \fi%
 \fi
 }%
}

\usesmartdiagramlibrary{core.definitions}
\usesmartdiagramlibrary{core.styles}
\usesmartdiagramlibrary{core.commands}
```

8.2 Keys and color declaration

The predefined colors:

```
\@namedef{color@1}{red!40}
\@namedef{color@2}{cyan!40}
\@namedef{color@3}{blue!40}
\@namedef{color@4}{green!40}
\@namedef{color@5}{orange!40}
\@namedef{color@6}{yellow!40}
\@namedef{color@7}{magenta!40}
\@namedef{color@8}{brown!40}
\@namedef{color@9}{violet!40}
\@namedef{color@10}{teal!40}
```

Basic shape definition and function to compute the height of the priority descriptive diagram:

```
\tikzset{rnd rectangle/.style={rectangle,rounded corners}
}

\def\CalcHeight(#1,#2)#3{%
\pgfpointdiff{\pgfpointanchor{#1}{south west}}{\pgfpointanchor{#2}{north west}}
\pgfmathsetmacro{\myheight}{veclen(\pgf@x,\pgf@y)}
\global\expandafter\edef\csname #3\endcsname{\myheight}
}
```

The key definition and the functions to set them:

```
\pgfkeys{/smart diagram/.cd, module minimum width/.initial=2cm,
   module minimum height/.initial={1cm},
   module y sep/.initial={1.65},
   module x sep/.initial={2.75},
   descriptive items y sep/.initial={1.75},
   text width/.initial={1.5cm},
   description title width/.initial={1.5cm},
   description text width/.initial={5cm},
   description title text width/.initial={1.25cm},
   description title font/.initial={\small},
   description font/.initial={\small},
```

```
74   description width/.initial={5.5cm},
75   font/.initial={\small},
76   border color/.initial={gray},
77   circular distance/.initial={2.75cm},
78   arrow line width/.initial={0.1cm},
79   module shape/.initial={rnd rectangle},
80   insert decoration/.initial={},
81   arrow tip/.initial={stealth},
82   arrow color/.initial={gray},
83   bubble center node size/.initial={4cm},
84   bubble center node font/.initial={\large},
85   bubble center node color/.initial={lightgray!60},
86   distance center/other bubbles/.initial=0.8cm,
87   distance text center bubble/.initial={0.5cm},
88   bubble fill opacity/.initial={0.5},
89   bubble node size/.initial={2.5cm},
90   bubble text opacity/.initial={0.8},
91   bubble node font/.initial={\normalfont},
92   planet size/.initial={2.5cm},
93   planet color/.initial={lightgray!60},
94   planet font/.initial={\large},
95   distance planet-connection/.initial={0.1cm},
96   distance planet-text/.initial={0.5cm},
97   planet text width/.initial={1.75cm},
98   satellite size/.initial={1.75cm},
99   satellite font/.initial={\normalfont},
100  satellite fill opacity/.initial={0.5},
101  satellite text opacity/.initial={0.8},
102  satellite text width/.initial={1.5cm},
103  distance satellite-connection/.initial={0.075cm},
104  connection line width/.initial={0.1cm},
105  connection color/.initial={gray},
106  distance planet-satellite/.initial={3.5cm},
107  priority arrow width/.initial={1.5cm},
108  priority arrow head extend/.initial={0.15cm},
109  priority tick size/.initial={5pt},
110  priority arrow height advance/.initial={2cm},
111  sequence item height/.initial={1cm},
112  sequence item width/.initial={2cm},
113  sequence item border color/.initial={gray},
114  sequence item border size/.initial={1.75\pgflinewidth},
115  sequence item font size/.initial={\normalfont},
116  sequence item fill opacity/.initial={1},
117  sequence item text opacity/.initial={1},
118  sequence item text width/.initial={1.9cm},
119  sequence item text color/.initial={black},
120  sequence item uniform color/.initial={gray!60!black},
121  arrow style/.initial={<-},
122  text color/.initial={black},
123  back arrow distance/.initial={0.5},
```

```
}%

\pgfkeys{/smart diagram/.cd, module minimum width/.get=\sm@core@modulewidth,
   module minimum height/.get=\sm@core@moduleheight,
   module y sep/.get=\sm@core@moduleysep,
   module x sep/.get=\sm@core@modulexsep,
   descriptive items y sep/.get=\sm@core@descriptiveitemsysep,
   text width/.get=\sm@core@moduletextwidth,
   description title width/.get=\sm@core@descriptiontitlewidth,
   description text width/.get=\sm@core@descriptiontextwidth,
   description title text width/.get=\sm@core@descriptiontitletextwidth,
   description title font/.get=\sm@core@descriptiontitlefontsize,
   description font/.get=\sm@core@descriptionfontsize,
   description width/.get=\sm@core@descriptionwidth,
   font/.get=\sm@core@modulefontsize,
   border color/.get=\sm@core@bordercolor,
   circular distance/.get=\sm@core@circulardistance,
   arrow line width/.get=\sm@core@arrowlinewidth,
   module shape/.get=\sm@core@moduleshape,
   insert decoration/.get=\sm@core@borderdecoration,
   arrow tip/.get=\sm@core@arrowtip,
   arrow color/.get=\sm@core@arrowcolor,
   bubble center node size/.get=\sm@core@bubblecenternodesize,
   bubble center node font/.get=\sm@core@bubblecenternodefont,
   bubble center node color/.get=\sm@core@bubblecenternodecolor,
   distance center/other bubbles/.get=\sm@core@distancecenterotherbubbles,
   distance text center bubble/.get=\sm@core@distancetextcenterbubble,
   bubble fill opacity/.get=\sm@core@bubblefillopacity,
   bubble node size/.get=\sm@core@bubblenodesize,
   bubble text opacity/.get=\sm@core@bubbletextopacity,
   bubble node font/.get=\sm@core@bubblenodefont,
   planet size/.get=\sm@core@planetminimumsize,
   planet color/.get=\sm@core@planetcolor,
   planet font/.get=\sm@core@planetfont,
   distance planet-connection/.get=\sm@core@planetoutersep,
   distance planet-text/.get=\sm@core@planetinnersep,
   planet text width/.get=\sm@core@planettextwidth,
   satellite size/.get=\sm@core@satelliteminimumsize,
   satellite font/.get=\sm@core@satellitefont,
   satellite fill opacity/.get=\sm@core@satellitefillopacity,
   satellite text opacity/.get=\sm@core@satellitetextopacity,
   satellite text width/.get=\sm@core@satellitetextwidth,
   distance satellite-connection/.get=\sm@core@satelliteoutersep,
   connection line width/.get=\sm@core@connectionlinewidth,
   connection color/.get=\sm@core@connectioncolor,
   distance planet-satellite/.get=\sm@core@distanceplanetsatellite,
   priority arrow width/.get=\sm@core@priorityarrowwidth,
   priority arrow head extend/.get=\sm@core@priorityarrowheadextend,
   priority tick size/.get=\sm@core@prioritytick,
   priority arrow height advance/.get=\sm@core@priorityarrowheightadvance,
```

```
    sequence item height/.get=\sm@core@seqitemheight,
    sequence item width/.get=\sm@core@seqitemwidth,
    sequence item border color/.get=\sm@core@seqitembordercolor,
    sequence item border size/.get=\sm@core@seqlinewidth,
    sequence item font size/.get=\sm@core@seqitemfont,
    sequence item fill opacity/.get=\sm@core@seqitemfillopacity,
    sequence item text opacity/.get=\sm@core@seqitemtextopacity,
    sequence item text width/.get=\sm@core@seqitemtextwidth,
    sequence item text color/.get=\sm@core@seqitemtextcolor,
    sequence item uniform color/.get=\sm@core@seqitemuniformcol,
    arrow style/.get=\sm@core@arrowstyle,
    text color/.get=\sm@core@textcolor,
    back arrow distance/.get=\sm@core@backarrowdistance,
}%
```

The specific key to set the list of colors:

```
\pgfkeys{/smart diagram/.cd, set color list/.code={%
      \foreach \listitem [count=\i] in {#1}{%
         \global\@namedef{color@\i\expandafter}\expandafter{\listitem}%
      }%
   }%
}

\pgfkeys{/smart diagram/.cd, uniform color list/.code args={#1 for #2 items}{%
      \foreach \listitem [count=\i] in {1,...,#2}{%
         \global\@namedef{color@\i\expandafter}\expandafter{#1}%
      }%
   }%
}

\pgfkeys{/smart diagram/.cd, use predefined color list/.code={%
      \@namedef{color@1}{red!30}%
      \@namedef{color@2}{cyan!30}%
      \@namedef{color@3}{blue!30}%
      \@namedef{color@4}{green!30}%
      \@namedef{color@5}{orange!30}%
      \@namedef{color@6}{yellow!30}%
      \@namedef{color@7}{magenta!30}%
      \@namedef{color@8}{brown!30}%
      \@namedef{color@9}{violet!30}%
      \@namedef{color@10}{teal!30}%
   }
}
```

The specific key to disable the back arrow in the `flow diagram` and in the `flow diagram:horizontal`:

```
\newif\ifbackarrowdisabled
\pgfkeys{/smart diagram/.cd,
  back arrow disabled/.is if=backarrowdisabled,
  back arrow disabled=false,
}
```

The specific key to disable the final arrow in the `circular diagram` and in the `circular diagram: clockwise`:

```
\newif\ifcircularfinalarrowdisabled
\pgfkeys{/smart diagram/.cd,
  circular final arrow disabled/.is if=circularfinalarrowdisabled,
  circular final arrow disabled=false,
}
```

The command to activate the various keys:

```
\NewDocumentCommand{\smartdiagramset}{m}{%
  \pgfkeys{/smart diagram/.cd,#1}%
  \pgfkeys{/smart diagram/.cd, module minimum width/.get=\sm@core@modulewidth,
   module minimum height/.get=\sm@core@moduleheight,
   module y sep/.get=\sm@core@moduleysep,
   module x sep/.get=\sm@core@modulexsep,
   descriptive items y sep/.get=\sm@core@descriptiveitemsysep,
   text width/.get=\sm@core@moduletextwidth,
   description title width/.get=\sm@core@descriptiontitlewidth,
   description text width/.get=\sm@core@descriptiontextwidth,
   description title text width/.get=\sm@core@descriptiontitletextwidth,
   description title font/.get=\sm@core@descriptiontitlefontsize,
   description font/.get=\sm@core@descriptionfontsize,
   description width/.get=\sm@core@descriptionwidth,
   font/.get=\sm@core@modulefontsize,
   border color/.get=\sm@core@bordercolor,
   circular distance/.get=\sm@core@circulardistance,
   arrow line width/.get=\sm@core@arrowlinewidth,
   module shape/.get=\sm@core@moduleshape,
   insert decoration/.get=\sm@core@borderdecoration,
   arrow tip/.get=\sm@core@arrowtip,
   arrow color/.get=\sm@core@arrowcolor,
   bubble center node size/.get=\sm@core@bubblecenternodesize,
   bubble center node font/.get=\sm@core@bubblecenternodefont,
   bubble center node color/.get=\sm@core@bubblecenternodecolor,
   distance center/other bubbles/.get=\sm@core@distancecenterotherbubbles,
   distance text center bubble/.get=\sm@core@distancetextcenterbubble,
   bubble fill opacity/.get=\sm@core@bubblefillopacity,
   bubble node size/.get=\sm@core@bubblenodesize,
   bubble text opacity/.get=\sm@core@bubbletextopacity,
   bubble node font/.get=\sm@core@bubblenodefont,
   planet size/.get=\sm@core@planetminimumsize,
   planet color/.get=\sm@core@planetcolor,
   planet font/.get=\sm@core@planetfont,
   distance planet-connection/.get=\sm@core@planetoutersep,
   distance planet-text/.get=\sm@core@planetinnersep,
   planet text width/.get=\sm@core@planettextwidth,
   satellite size/.get=\sm@core@satelliteminimumsize,
   satellite font/.get=\sm@core@satellitefont,
   satellite fill opacity/.get=\sm@core@satellitefillopacity,
   satellite text opacity/.get=\sm@core@satellitetextopacity,
```

```
   satellite text width/.get=\sm@core@satellitetextwidth,
   distance satellite-connection/.get=\sm@core@satelliteoutersep,
   connection line width/.get=\sm@core@connectionlinewidth,
   connection color/.get=\sm@core@connectioncolor,
   distance planet-satellite/.get=\sm@core@distanceplanetsatellite,
   priority arrow width/.get=\sm@core@priorityarrowwidth,
   priority arrow head extend/.get=\sm@core@priorityarrowheadextend,
   priority tick size/.get=\sm@core@prioritytick,
   priority arrow height advance/.get=\sm@core@priorityarrowheightadvance,
   sequence item height/.get=\sm@core@seqitemheight,
   sequence item width/.get=\sm@core@seqitemwidth,
   sequence item border color/.get=\sm@core@seqitembordercolor,
   sequence item border size/.get=\sm@core@seqlinewidth,
   sequence item font size/.get=\sm@core@seqitemfont,
   sequence item fill opacity/.get=\sm@core@seqitemfillopacity,
   sequence item text opacity/.get=\sm@core@seqitemtextopacity,
   sequence item text width/.get=\sm@core@seqitemtextwidth,
   sequence item text color/.get=\sm@core@seqitemtextcolor,
   sequence item uniform color/.get=\sm@core@seqitemuniformcol,
   arrow style/.get=\sm@core@arrowstyle,
   text color/.get=\sm@core@textcolor,
   back arrow distance/.get=\sm@core@backarrowdistance,
  }%
}%
```

Key to let the sequence color be uniform:

```
\pgfkeys{/smart diagram/.cd,%
    uniform sequence color/.is choice,%
    uniform sequence color/true/.code={%
      \tikzset{sequence item/.append style={%
         fill=\sm@core@seqitemuniformcol,
         },%
       }%
    },%
    uniform sequence color/false/.style={sequence item},%
    uniform sequence color/.default=false,%
}%
```

Key to let the connection planet satellite color be uniform:

```
\pgfkeys{/smart diagram/.cd,%
      uniform connection color/.is choice,%
      uniform connection color/true/.code={%
        \tikzset{connection planet satellite/.append style={%
           \sm@core@connectioncolor
           },%
        }%
      },%
      uniform connection color/false/.style={connection planet satellite},%
    uniform connection color/.default=false,%
}%
```

Key to let the arrow color be uniform:

```
\pgfkeys{/smart diagram/.cd,%
      uniform arrow color/.is choice,%
      uniform arrow color/true/.code={%
        \tikzset{diagram arrow type/.append style={%
           \sm@core@arrowcolor
           },%
        }%
      },%
      uniform arrow color/false/.style={diagram arrow type},%
    uniform arrow color/.default=false,%
}%

```

The fading style applied to the priority descriptive diagram and styles diagram definition:

```
\tikzfading[name=priorityarrowfading,
  bottom color=transparent!5,
  top color=transparent!80
]
\tikzset{priority arrow fill/.style={
  fill=gray,
  path fading=priorityarrowfading
  }
}

\tikzset{module/.style={%
      \pgfkeysvalueof{/smart diagram/module shape},
      thick,
      draw=\sm@core@bordercolor,
      top color=white,
      bottom color=\col,
      text=\sm@core@textcolor,
      text width=\sm@core@moduletextwidth,
      minimum width=\sm@core@modulewidth,
      minimum height=\sm@core@moduleheight,
      font=\sm@core@modulefontsize,
      \sm@core@borderdecoration
   },
   diagram arrow type/.style={%
      \sm@core@arrowstyle,
      >=\sm@core@arrowtip,
      line width=\sm@core@arrowlinewidth,
      \col
   },%
}
\tikzset{description title/.style={%
      circle,
      draw=\sm@core@bordercolor,
      minimum width=\sm@core@descriptiontitlewidth,
```

```
      anchor=east,
      bottom color=\col,
      top color=white!80!\col,
      font=\sm@core@descriptiontitlefontsize,
      text width=\sm@core@descriptiontitletextwidth,
      \sm@core@borderdecoration,
   },
   description/.style={%
      \pgfkeysvalueof{/smart diagram/module shape},
      text width=\sm@core@descriptiontextwidth,
      draw=\sm@core@bordercolor,
      anchor=west,
      minimum height=\sm@core@moduleheight,
      minimum width=\sm@core@descriptionwidth,
      bottom color=\col,
      top color=white!80!\col,
      font=\sm@core@descriptionfontsize,
      \sm@core@borderdecoration,
   }%
}
\tikzset{priority arrow/.style={
      draw=\sm@core@bordercolor,
      single arrow,
      minimum height=\distancemodules,
      minimum width=\sm@core@priorityarrowwidth,
      priority arrow fill,
      rotate=90,
      single arrow head extend=\sm@core@priorityarrowheadextend,
      anchor=west,
   }
}
\tikzset{bubble center node/.style={
      minimum size=\sm@core@bubblecenternodesize,
      circle,
      fill=\sm@core@bubblecenternodecolor,
      font=\sm@core@bubblecenternodefont,
      outer sep=\sm@core@distancecenterotherbubbles,
      inner sep=\sm@core@distancetextcenterbubble,
   },
   bubble node/.style={
      minimum size=\sm@core@bubblenodesize,
      circle,
   ultra thick,
   font=\sm@core@bubblenodefont,
   draw=white,
   fill opacity=\sm@core@bubblefillopacity,
   fill=\col,
   text opacity=\sm@core@bubbletextopacity,
   }
}
```

```
\tikzset{planet/.style={
      minimum size=\sm@core@planetminimumsize,
      circle,
      fill=\sm@core@planetcolor,
      font=\sm@core@planetfont,
      outer sep=\sm@core@planetoutersep,
      inner sep=\sm@core@planetinnersep,
      text width=\sm@core@planettextwidth,
   },
   satellite/.style={
      minimum size=\sm@core@satelliteminimumsize,
      circle,
    font=\sm@core@satellitefont,
    fill opacity=\sm@core@satellitefillopacity,
    fill=\col,
    text opacity=\sm@core@satellitetextopacity,
    text width=\sm@core@satellitetextwidth,
    outer sep=\sm@core@satelliteoutersep,
   },
   connection planet satellite/.style={
     ->,
     line width=\sm@core@connectionlinewidth,
     >=\sm@core@arrowtip,
     \col,
   }
}

\tikzset{sequence item/.style={
      minimum height=\sm@core@seqitemheight,
      minimum width=\sm@core@seqitemwidth,
      signal,
      signal from=west,
      signal to=east,
      draw=\sm@core@seqitembordercolor,
      line width=\sm@core@seqlinewidth,
    font=\sm@core@seqitemfont,
    fill opacity=\sm@core@seqitemfillopacity,
    fill=\col,
    text opacity=\sm@core@seqitemtextopacity,
    text width=\sm@core@seqitemtextwidth,
      text=\sm@core@seqitemtextcolor,
  }
}
% let the first word of the item be hypenate
\tikzset{let hypenation/.style={%
    execute at begin node={%
       \hspace{0pt}%
    }%
  }%
}%
```

The definition of the visibility style:

```
\tikzset{%
   sminvisible/.style={opacity=0,text opacity=0},
   smvisible on/.style={smalt=#1{}{sminvisible}},
   smalt/.code args={<#1>#2#3}{%
      \alt<#1>{\pgfkeysalso{#2}}{\pgfkeysalso{#3}}
   },%
}%
```

8.3 Commands

Definition of the two commands. The diagrams:

```
\NewDocumentCommand{\smartdiagram}{r[] m}{%
   \StrCut{#1}{:}\diagramtype\option
   \IfNoValueTF{#1}{% true-no value 1
      \PackageError{smartdiagram}%
      {Type of the diagram not inserted. Please insert it}%
      {Example: \protect\smartdiagram[flow diagram]}}
   {%false-no value 1
   \IfStrEq{\diagramtype}{}{%
      \PackageError{smartdiagram}{Type of the diagram not inserted. Please insert it}
      {Example: \protect\smartdiagram[flow diagram]}
   }{}
   \IfStrEq{\diagramtype}{circular diagram}{% true-circular diagram
   \begin{tikzpicture}[every node/.style={align=center,let hypenation}]

   \foreach \smitem [count=\xi] in {#2}  {\global\let\maxsmitem\xi}

   \foreach \smitem [count=\xi] in {#2}{%
   \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
     \pgfmathtruncatemacro{\angle}{180+360/\maxsmitem*\xi}
   }{% false-clockwise-circular diagram
     \pgfmathtruncatemacro{\angle}{360/\maxsmitem*\xi}
   }
   \edef\col{\@nameuse{color@\xi}}
   \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
     \node[module,drop shadow] (module\xi)
      at (-\angle:\sm@core@circulardistance) {\smitem };
   }{% false-clockwise-circular diagram
     \node[module,drop shadow] (module\xi)
      at (\angle:\sm@core@circulardistance) {\smitem };
   }
   }%
   \foreach \smitem [count=\xi] in {#2}{%
   \ifnum\xi=\maxsmitem
     \ifcircularfinalarrowdisabled
       \relax
     \else
       \pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
```

```
      \edef\col{\@nameuse{color@\xj}}
      \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
        \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm]
         (module\xj) to[bend right] (module\xi);
        }{% false-clockwise-circular diagram
         \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm]
          (module\xj) to[bend left] (module\xi);
        }
    \fi
  \else
    \pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
    \edef\col{\@nameuse{color@\xj}}
    \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
      \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm]
        (module\xj) to[bend right] (module\xi);
    }{% false-clockwise-circular diagram
       \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm]
        (module\xj) to[bend left] (module\xi);
      }

  \fi
  }%
  \end{tikzpicture}
  }{}% end-circular diagram
  \IfStrEq{\diagramtype}{flow diagram}{% true-flow diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]

  \foreach \smitem [count=\xi] in {#2}  {\global\let\maxsmitem\xi}

  \foreach \smitem [count=\xi] in {#2}{%
  \edef\col{\@nameuse{color@\xi}}
  \IfStrEq{\option}{horizontal}{% true-horizontal-flow diagram
    \path let \n1 = {int(0-\xi)}, \n2={0+\xi*\sm@core@modulexsep} in
        node[module,drop shadow] (module\xi) at +(\n2,0) {\smitem};
  }{% false-horizontal-flow diagram
    \path let \n1 = {int(0-\xi)}, \n2={0-\xi*\sm@core@moduleysep} in
        node[module,drop shadow] (module\xi) at +(0,\n2) {\smitem};
  }
  }%

  \foreach \smitem [count=\xi] in {#2}{%
  \pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
  \edef\col{\@nameuse{color@\xj}}
  \ifnum\xi<\maxsmitem
  \begin{pgfonlayer}{smart diagram arrow back}
  \draw[diagram arrow type] (module\xj) -- (module\xi);
  \end{pgfonlayer}
  \fi
  % last arrow - not display it in background - check if disabled
  \ifbackarrowdisabled
```

```
    \relax
  \else
    \ifnum\xi=\maxsmitem
      \IfStrEq{\option}{horizontal}{% true-horizontal-flow diagram
        \tikzset{square arrow/.style={%
           to path={-- ++(0,\sm@core@backarrowdistance) -| (\tikztotarget)}
           }%
        }%
        \draw[diagram arrow type, square arrow]
         (module\xj.north) to (module\xi.north);
      }{% false-horizontal-flow diagram
        \tikzset{square arrow/.style={%
           to path={-- ++(\sm@core@backarrowdistance,0) |- (\tikztotarget)}
           }%
        }%
        \draw[diagram arrow type,square arrow]
          (module\xj.east) to (module\xi);
      }%
    \fi
  \fi
  }%
  \end{tikzpicture}
  }{}% end-flow diagram
  \IfStrEq{\diagramtype}{descriptive diagram}{% true-descr. diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{%
  \edef\col{\@nameuse{color@\xi}}
  \foreach \subitem [count=\xii] in \smitem{%
  \ifnumequal{\xii}{1}{% true
  \node[description title,drop shadow]
   (module-title\xi) at (0,0-\xi*\sm@core@descriptiveitemsysep) {\subitem};
  }{}
  \ifnumequal{\xii}{2}{% true
  \node[description,drop shadow](module\xi)
   at (0,0-\xi*\sm@core@descriptiveitemsysep) {\subitem};
  }{}
  }%
  }%
  \end{tikzpicture}
  }{}% end-descr. diagram
  \IfStrEq{\diagramtype}{bubble diagram}{% true-bubble diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
  \pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
  \foreach \smitem [count=\xi] in {#2}{%
  \ifnumequal{\xi}{1}{ %true
  \node[bubble center node](center bubble){\smitem};
  }{%false
  \pgfmathtruncatemacro{\xj}{\xi-1}
  \pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
```

```
\edef\col{\@nameuse{color@\xj}}
\node[bubble node] (module\xi)
    at (center bubble.\angle) {\smitem };
}%
}%
\end{tikzpicture}
}{}%end-bubble diagram
\IfStrEq{\diagramtype}{constellation diagram}{% true-const diagram
\begin{tikzpicture}[every node/.style={align=center,let hypenation}]
\foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
\pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
\foreach \smitem [count=\xi] in {#2}{%
\ifnumequal{\xi}{1}{ %true
\node[planet](planet){\smitem};
}{%false
\pgfmathtruncatemacro{\xj}{\xi-1}
\pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
\edef\col{\@nameuse{color@\xj}}
\node[satellite] (satellite\xi)
    at (\angle:\sm@core@distanceplanetsatellite) {\smitem };
\draw[connection planet satellite] (planet) -- (satellite\xi);
}%
}%
\end{tikzpicture}
}{}%end-const diagram
\IfStrEq{\diagramtype}{connected constellation diagram}{% true-conn const diagram
\begin{tikzpicture}[every node/.style={align=center,let hypenation}]
\foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
\pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
\foreach \smitem [count=\xi] in {#2}{%
\ifnumequal{\xi}{1}{ %true
\node[planet](planet){\smitem};
}{%false
\pgfmathtruncatemacro{\xj}{\xi-1}
\pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
\edef\col{\@nameuse{color@\xj}}
\node[satellite] (satellite\xj)
     at (\angle:\sm@core@distanceplanetsatellite) {\smitem };
}%
}%
\foreach \smitem [count=\xi] in {#2}{%
   \ifnumgreater{\xi}{1}{ %true
   \pgfmathtruncatemacro{\xj}{\xi-1}
   \edef\col{\@nameuse{color@\xj}}
   \pgfmathtruncatemacro{\xk}{mod(\xj,\actualnumitem) +1}
   \path[connection planet satellite,-]
      (satellite\xj) edge[bend right] (satellite\xk);
}{}
}%
\end{tikzpicture}
```

```
   }{}%end-connected constellation diagram
   \IfStrEq{\diagramtype}{priority descriptive diagram}{% true-priority descriptive diagram
   \pgfmathparse{subtract(\sm@core@priorityarrowwidth,\sm@core@priorityarrowheadextend)}
   \pgfmathsetmacro\sm@core@prioritytricksize{\pgfmathresult/2}
   \pgfmathsetmacro\arrowtickxshift{(\sm@core@priorityarrowwidth-\sm@core@prioritytricksize)/2}
   \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
   \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
   \foreach \smitem [count=\xi] in {#2}{%
   \edef\col{\@nameuse{color@\xi}}
    \node[description,drop shadow](module\xi)
    at (0,0+\xi*\sm@core@descriptiveitemsysep) {\smitem};
\draw[line width=\sm@core@prioritytick,\col]
      ([xshift=-\arrowtickxshift pt]module\xi.base west)--
 ($([xshift=-\arrowtickxshift pt]module\xi.base west)-(\sm@core@prioritytricksize pt,0)$);
   }%
   \coordinate (A) at (module1);
   \coordinate (B) at (module\maxsmitem);
   \CalcHeight(A,B){heightmodules}
   \pgfmathadd{\heightmodules}{\sm@core@priorityarrowheightadvance}
   \pgfmathsetmacro{\distancemodules}{\pgfmathresult}
   \pgfmathsetmacro\arrowxshift{\sm@core@priorityarrowwidth/2}
   \begin{pgfonlayer}{background}
   \node[priority arrow] at ([xshift=-\arrowxshift pt]module1.south west){};
   \end{pgfonlayer}
   \end{tikzpicture}
   }{}% end-priority descriptive diagram
   \IfStrEq{\diagramtype}{sequence diagram}{% true-sequence diagram
   \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
   \foreach \x[count=\xi, count=\prevx from 0] in {#2}{%
   \edef\col{\@nameuse{color@\xi}}
   \ifnum\xi=1
     \node[sequence item] (sequence-item\xi) {\x};
   \else
     \node[sequence item,anchor=west] (sequence-item\xi) at (sequence-item\prevx.east) {\x};
   \fi
   }
   \end{tikzpicture}
   }{}% end-sequence diagram
}% end-no value 1
}% end-command
```

The command definition for the animated diagrams:

```
\NewDocumentCommand{\smartdiagramanimated}{r[] m}{%
   \StrCut{#1}{:}\diagramtype\option
   \IfNoValueTF{#1}{% true-no value 1
      \PackageError{smartdiagram}{Type of the diagram not inserted. Please insert it}
      {Example: \protect\smartdiagram[flow diagram]}}
   {%false-no value 1
   \IfStrEq{\diagramtype}{}{%
      \PackageError{smartdiagram}{Type of the diagram not inserted. Please insert it}
```

```
    {Example: \protect\smartdiagram[flow diagram]}
}{}
\IfStrEq{\diagramtype}{circular diagram}{% true-circular diagram
\begin{tikzpicture}[every node/.style={align=center,let hypenation}]
\foreach \smitem [count=\xi] in {#2}  {\global\let\maxsmitem\xi}
\foreach \smitem [count=\xi] in {#2}{%
\IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
  \pgfmathtruncatemacro{\angle}{180+360/\maxsmitem*\xi}
}{% false-clockwise-circular diagram
  \pgfmathtruncatemacro{\angle}{360/\maxsmitem*\xi}
}
\edef\col{\@nameuse{color@\xi}}
\IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
   \node[module,
     drop shadow={smvisible on=<\xi->},
     smvisible on=<\xi->] (module\xi)
    at (-\angle:\sm@core@circulardistance) {\smitem};
}{% false-clockwise-circular diagram
   \node[module,
     drop shadow={smvisible on=<\xi->},
     smvisible on=<\xi->] (module\xi)
    at (\angle:\sm@core@circulardistance) {\smitem};
}
}%
\foreach \smitem [count=\xi] in {#2}{%
\ifnum\xi=\maxsmitem
  \ifcircularfinalarrowdisabled
    \relax
  \else
    \pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
    \pgfmathtruncatemacro{\adv}{\xi + 1)}
    \edef\col{\@nameuse{color@\xj}}
    \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
      \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm,
     smvisible on=<\adv->]
       (module\xj) to[bend right] (module\xi);
      }{% false-clockwise-circular diagram
       \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm,
     smvisible on=<\adv->]
        (module\xj) to[bend left] (module\xi);
      }
  \fi
\else
  \pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
  \pgfmathtruncatemacro{\adv}{\xi + 1)}
  \edef\col{\@nameuse{color@\xj}}
  \IfStrEq{\option}{clockwise}{% true-clockwise-circular diagram
    \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm,
     smvisible on=<\adv->]
      (module\xj) to[bend right] (module\xi);
```

```
    }{% false-clockwise-circular diagram
       \draw[diagram arrow type,shorten <=0.3cm,shorten >=0.3cm,
         smvisible on=<\adv->]
        (module\xj) to[bend left] (module\xi);
      }

\fi
}%
\end{tikzpicture}
}{}% end-circular diagram
\IfStrEq{\diagramtype}{flow diagram}{% true-flow diagram
\begin{tikzpicture}[every node/.style={align=center,let hypenation}]

\foreach \smitem [count=\xi] in {#2}  {\global\let\maxsmitem\xi}

\foreach \smitem [count=\xi] in {#2}{%
\edef\col{\@nameuse{color@\xi}}
\IfStrEq{\option}{horizontal}{% true-horizontal-flow diagram
  \path let \n1 = {int(0-\xi)}, \n2={0+\xi*\sm@core@modulexsep}
     in node[module,drop shadow={smvisible on=<\xi->},
     smvisible on=<\xi->] (module\xi) at +(\n2,0) {\smitem};
}{% false-horizontal-flow diagram
  \path let \n1 = {int(0-\xi)}, \n2={0-\xi*\sm@core@moduleysep}
     in node[module,drop shadow={smvisible on=<\xi->},
     smvisible on=<\xi->] (module\xi) at +(0,\n2) {\smitem};
}
}%

\foreach \smitem [count=\xi] in {#2}{%
\pgfmathtruncatemacro{\xj}{mod(\xi, \maxsmitem) + 1)}
\edef\col{\@nameuse{color@\xj}}
\ifnum\xi<\maxsmitem
\begin{pgfonlayer}{smart diagram arrow back}
\draw[diagram arrow type,smvisible on=<\xi->]
  (module\xj) -- (module\xi);
\end{pgfonlayer}
\fi
% last arrow - not display it in background - check if disabled
\ifbackarrowdisabled
  \relax
\else
  \ifnum\xi=\maxsmitem
    \IfStrEq{\option}{horizontal}{% true-horizontal-flow diagram
      \tikzset{square arrow/.style={
        to path={-- ++(0,\sm@core@backarrowdistance) -| (\tikztotarget)}
        }
      }
      \draw[diagram arrow type, square arrow,smvisible on=<\xi->]
       (module\xj.north) to (module\xi.north);
    }{% false-horizontal-flow diagram
```

```
        \tikzset{square arrow/.style={
          to path={-- ++(\sm@core@backarrowdistance,0) |- (\tikztotarget)}
          }
        }
        \draw[diagram arrow type,square arrow,smvisible on=<\xi->]
          (module\xj.east) to (module\xi);
      }
    \fi
  \fi
  }%
  \end{tikzpicture}
  }{}% end-flow diagram
  \IfStrEq{\diagramtype}{descriptive diagram}{% true-descriptive diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{%
  \edef\col{\@nameuse{color@\xi}}

  \foreach \subitem [count=\xii] in \smitem{%
     \pgfmathtruncatemacro\subitemvisible{\xi}
  \ifnumequal{\xii}{1}{% true
  \node[description title,drop shadow, smvisible on=<\subitemvisible->]
  (module-title\xi) at (0,0-\xi*\sm@core@descriptiveitemsysep) {\subitem};\pause
  }{}
  \ifnumequal{\xii}{2}{% true
  \node[description,drop shadow,smvisible on=<\subitemvisible->]
  (module\xi)at (0,0-\xi*\sm@core@descriptiveitemsysep) {\subitem};\pause
  }{}
  }%
  }%
  \end{tikzpicture}
  }{}% end-descriptive diagram
  \IfStrEq{\diagramtype}{bubble diagram}{% true-bubble diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
  \pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
  \foreach \smitem [count=\xi] in {#2}{%
  \ifnumequal{\xi}{1}{ %true
  \node[bubble center node, smvisible on=<\xi->](center bubble){\smitem};
  }{%false
  \pgfmathtruncatemacro{\xj}{\xi-1}
  \pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
  \edef\col{\@nameuse{color@\xj}}
  \node[bubble node, smvisible on=<\xi->](module\xi)
       at (center bubble.\angle) {\smitem };
  }%
  }%
  \end{tikzpicture}
  }{}%end-bubble diagram
  \IfStrEq{\diagramtype}{constellation diagram}{% true-const diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
```

```
  \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
  \pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
  \foreach \smitem [count=\xi] in {#2}{%
  \ifnumequal{\xi}{1}{ %true
  \node[planet, smvisible on=<\xi->](planet){\smitem};
  }{%false
  \pgfmathtruncatemacro{\xj}{\xi-1}
  \pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
  \edef\col{\@nameuse{color@\xj}}
  \node[satellite, smvisible on=<\xi->] (satellite\xi)
   at (\angle:\sm@core@distanceplanetsatellite) {\smitem };
  \draw[connection planet satellite, smvisible on=<\xi->]
   (planet) -- (satellite\xi);
}%
  }%
  \end{tikzpicture}
  }{}%end-constellation diagram
  \IfStrEq{\diagramtype}{connected constellation diagram}{% true-conn const diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
  \pgfmathtruncatemacro\actualnumitem{\maxsmitem-1}
  \foreach \smitem [count=\xi] in {#2}{%
  \ifnumequal{\xi}{1}{ %true
  \node[planet,smvisible on=<\xi->](planet){\smitem};
  }{%false
  \pgfmathtruncatemacro{\xj}{\xi-1}
  \pgfmathtruncatemacro{\angle}{360/\actualnumitem*\xj}
  \edef\col{\@nameuse{color@\xj}}
  \node[satellite,smvisible on=<\xi->] (satellite\xj)
   at (\angle:\sm@core@distanceplanetsatellite) {\smitem };
  }%
  }%
  \foreach \smitem [count=\xi] in {#2}{%
     \ifnumgreater{\xi}{1}{ %true
     \pgfmathtruncatemacro{\xj}{\xi-1}
     \edef\col{\@nameuse{color@\xj}}
     \pgfmathtruncatemacro{\xk}{mod(\xj,\actualnumitem) +1}
     \pgfmathtruncatemacro{\smvisible}{\xi+1}
     \path[connection planet satellite,-,smvisible on=<\smvisible->]
      (satellite\xj) edge[bend right] (satellite\xk);
  }{}
  }%
  \end{tikzpicture}
  }{}%end-connected constellation diagram
  \IfStrEq{\diagramtype}{priority descriptive diagram}{% true-priority descriptive diagram
  \pgfmathparse{subtract(\sm@core@priorityarrowwidth,\sm@core@priorityarrowheadextend)}
  \pgfmathsetmacro\sm@core@prioritytricksize{\pgfmathresult/2}
  \pgfmathsetmacro\arrowtickxshift{(\sm@core@priorityarrowwidth-\sm@core@prioritytricksize)/2}
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \smitem [count=\xi] in {#2}{\global\let\maxsmitem\xi}
```

```
  \foreach \smitem [count=\xi] in {#2}{%
  \edef\col{\@nameuse{color@\xi}}
  \pgfmathtruncatemacro\smvisible{\xi+1}
  \node[description,drop shadow={smvisible on=<\smvisible->},smvisible on=<\smvisible->]
   (module\xi) at (0,0+\xi*\sm@core@descriptiveitemsysep) {\smitem};
\draw[line width=\sm@core@prioritytick,\col,smvisible on=<\smvisible->]
 ([xshift=-\arrowtickxshift pt]module\xi.base west)--
 ($([xshift=-\arrowtickxshift pt]module\xi.base west)-(\sm@core@prioritytick size pt,0)$);
  }%
  \coordinate (A) at (module1);
  \coordinate (B) at (module\maxsmitem);
  \CalcHeight(A,B){heightmodules}
  \pgfmathadd{\heightmodules}{\sm@core@priorityarrowheightadvance}
  \pgfmathsetmacro{\distancemodules}{\pgfmathresult}
  \pgfmathsetmacro\arrowxshift{\sm@core@priorityarrowwidth/2}
  \begin{pgfonlayer}{background}
  \node[priority arrow] at ([xshift=-\arrowxshift pt]module1.south west){};
  \end{pgfonlayer}
  \end{tikzpicture}
  }{}% end-priority descriptive diagram
  \IfStrEq{\diagramtype}{sequence diagram}{% true-sequence diagram
  \begin{tikzpicture}[every node/.style={align=center,let hypenation}]
  \foreach \x[count=\xi, count=\prevx from 0] in {#2}{%
  \edef\col{\@nameuse{color@\xi}}
  \ifnum\xi=1
    \node[sequence item,smvisible on=<\xi->] (sequence-item\xi) {\x};
  \else
    \node[sequence item,anchor=west,smvisible on=<\xi->]
     (sequence-item\xi) at (sequence-item\prevx.east) {\x};
  \fi
  }
  \end{tikzpicture}
  }{}% end-sequence diagram
  }% end-no value 1
}% end-command
```

8.4 Library Additions

The library at first loads the Ti*k*Z library positioning.

```
\usetikzlibrary{positioning}
```

Key definition:

```
\pgfkeys{/smart diagram/additions/.cd,
  additional item shape/.initial=\pgfkeysvalueof{/smart diagram/module shape},
  additional item shape/.get=\sm@additions@additionalitemshape,
  additional item shape/.store in=\sm@additions@additionalitemshape,
  additional item border color/.initial=none,
  additional item border color/.get=\sm@additions@additionalitembordercolor,
  additional item border color/.store in=\sm@additions@additionalitembordercolor,
  additional item bottom color/.initial=white,
```

```
  additional item bottom color/.get=\sm@additions@additionalitembottomcolor,
  additional item bottom color/.store in=\sm@additions@additionalitembottomcolor,
  additional item fill color/.initial=none,
  additional item fill color/.get=\sm@additions@additionalitemfillcolor,
  additional item fill color/.store in=\sm@additions@additionalitemfillcolor,
  additional item text width/.initial=1.75cm,
  additional item text width/.get=\sm@additions@additionalitemtextwidth,
  additional item text width/.store in=\sm@additions@additionalitemtextwidth,
  additional item width/.initial=2cm,
  additional item width/.get=\sm@additions@additionalitemwidth,
  additional item width/.store in=\sm@additions@additionalitemwidth,
  additional item height/.initial=1cm,
  additional item height/.get=\sm@additions@additionalitemheight,
  additional item height/.store in=\sm@additions@additionalitemheight,
  additional item font/.initial=\normalfont,
  additional item font/.get=\sm@additions@additionalitemfont,
  additional item font/.store in=\sm@additions@additionalitemfont,
  additional item border decoration/.initial={},
  additional item border decoration/.get=\sm@additions@additionalitemdecoration,
  additional item border decoration/.store in=\sm@additions@additionalitemdecoration,
  additional item offset/.initial={0.25cm},
  additional item offset/.get=\sm@additions@additionalitemoffset,
  additional item offset/.store in=\sm@additions@additionalitemoffset,
  additional item fill opacity/.initial={1},
  additional item fill opacity/.get=\sm@additions@additionalitemfillopacity,
  additional item fill opacity/.store in=\sm@additions@additionalitemfillopacity,
  additional item text opacity/.initial={1},
  additional item text opacity/.get=\sm@additions@additionalitemtextopacity,
  additional item text opacity/.store in=\sm@additions@additionalitemtextopacity,
  additional arrow tip/.initial={stealth},
  additional arrow tip/.get=\sm@additions@additionalarrowtip,
  additional arrow tip/.store in=\sm@additions@additionalarrowtip,
  additional arrow line width/.initial={0.1cm},
  additional arrow line width/.get=\sm@additions@additionalarrowlinewidth,
  additional arrow line width/.store in=\sm@additions@additionalarrowlinewidth,
  additional arrow color/.initial={gray},
  additional arrow color/.get=\sm@additions@additionalarrowcolor,
  additional arrow color/.store in=\sm@additions@additionalarrowcolor,
  additional arrow style/.initial={->},
  additional arrow style/.get=\sm@additions@additionalarrowstyle,
  additional arrow style/.store in=\sm@additions@additionalarrowstyle,
  additional item shadow/.initial={},
  additional item shadow/.get=\sm@additions@additionalitemshadow,
  additional item shadow/.store in=\sm@additions@additionalitemshadow,
}

\newif\ifconnectionsdisabled
\pgfkeys{/smart diagram/additions/.cd,
  additional connections disabled/.is if=connectionsdisabled,
  additional connections disabled=true,
```

```
}

\pgfkeys{/smart diagram/.cd,
  additions/.style={/smart diagram/additions/.cd,#1}%
}
```

Style definition; the `additional item` style comprises lot of usual Ti*k*Z options: it possible to select a coloring with a vertical shading or an uniform filling.

```
\tikzset{additional item/.style={
      align=center,
      \sm@additions@additionalitemshape,
      thick,
      draw=\sm@additions@additionalitembordercolor,
      top color=white,
      bottom color=\sm@additions@additionalitembottomcolor,
      postaction={fill=\sm@additions@additionalitemfillcolor},
      text width=\sm@additions@additionalitemtextwidth,
      minimum width=\sm@additions@additionalitemwidth,
      minimum height=\sm@additions@additionalitemheight,
      font=\sm@additions@additionalitemfont,
      fill opacity=\sm@additions@additionalitemfillopacity,
      text opacity=\sm@additions@additionalitemtextopacity,
      \sm@additions@additionalitemshadow,
      \sm@additions@additionalitemdecoration
   },
   additional item arrow type/.style={
      \sm@additions@additionalarrowstyle,
      >=\sm@additions@additionalarrowtip,
      line width=\sm@additions@additionalarrowlinewidth,
      \sm@additions@additionalarrowcolor
   },
}
```

Command definition; at first the diagram is created with the usual command, then the foreach iterates in order to get additions. The additions' strings are cut by means of the package xstring and its macro `\StrCut`. Of course, to all the tikzpictures, the option `remember picture` is added.

```
\NewDocumentCommand{\smartdiagramadd}{r[] m m}{
\tikzstyle{every picture}+=[remember picture]
\smartdiagram[#1]{#2}
\begin{tikzpicture}[remember picture,overlay,
  every node/.style={align=center,let hypenation}]
\foreach \smitem [count=\xi] in {#2}  {\global\let\numitems\xi}
\foreach \smitem[count=\xi] in {#3}{
\StrCut{\smitem}{/}\pos\textitem
\StrCut{\pos}{\space of\space}\point\modulenum
 \node[additional item,
   \point=\sm@additions@additionalitemoffset of \modulenum]
  (additional-module\xi)  {\textitem};
 \ifconnectionsdisabled
   \relax
```

```
 \else
  \begin{pgfonlayer}{smart diagram arrow back}
   \draw[additional item arrow type]
    (additional-module\xi) -- (\modulenum);
  \end{pgfonlayer}
 \fi
}
\end{tikzpicture}
}
```

Definition of the command to connect additions with diagram modules:

```
\NewDocumentCommand{\smartdiagramconnect}{m m}{%
  \begin{tikzpicture}[remember picture,overlay]
  \foreach \start/\end in {#2}
  \draw[additional item arrow type,#1]
    (\start) -- (\end);
  \end{tikzpicture}
}
```