ARAKHNE.ORG
Another Yet Free Software Gateway

LaTeX Packages for Unified Process Methodology

LaTeX Packages for Unified Process Methodology

Official Documentation

Reference:	UPM-2012-01
Version:	28.4
Updated:	2013/10/09
Status:	Public

Authors: Stéphane Galland

This document describes the LaTeX Packages for Unified Process Methodology project.

TeX is a trademark of the American Mathematical Society.

`tex-upmethodology` is owned by Stéphane Galland, *Arakhnê*.org, France.

This document was realised with LaTeX and `tex-upmethodology`.

Reference : UPM-2012-01

Document Summary	
Project	LaTeX Packages for Unified Process Methodology
Document	Official Documentation
Reference	UPM-2012-01
Version	28.4
Last Update	2013/10/09

Authors		
Names	*Comments*	*Emails*
Stéphane Galland	Original Author	galland@arakhne.org

Validators			
Names	*Comments*	*Emails*	*Initials*
Stéphane Galland	Original Author	galland@arakhne.org	

Version History		
Version	*Date*	*Updates*
18.0	2012/12/21	\upmuse → \UseExtension
19.0	2013/04/28	Replace the package inclusion of `color.sty` by `xcolor.sty`
20.0	2013/05/17	Add a section on the default configuration of the package `graphicx`
21.0	2013/05/30	Add the macros `\includegraphicswtex` and `\DeclareGraphicsExtensionsWtex`
21.1	2013/05/31	Fixing some grammar and syntax problems.
21.2	2013/07/06	Add the macro `\ifdocumentauthor`.
22.0	2013/08/15	Add the macros: `\textup`, `\textsubscript`, `\textdown`, `\textsubscript`, `\Emph`. Add the environments `enumdescription`, `enumdescriptionx`.
23.0	2013/08/17	Update the second parameter of `\mfigurewtex` to have the same semantic as for `\mfigure`.
24.0	2013/08/26	Major bug fixes in the environments `mtabular` and `mtable`. Add the macro `\tabulartitlespec`. Add the macro `\makenamespacing`.
25.0	2013/08/27	Add the environment `framedcolorminipage`. Add the `definition` environment.
26.0	2013/09/02	Change the specification of the optional parameter of `mtable`. Add the macro `\makelastname`
27.0	2013/09/05	Add the macro `\declareupmtheorem`
28.0	2013/09/19	Add the mathematic symbols
28.1	2013/09/26	Add the missed table placement flag `p`
28.2	2013/09/29	Add a notification text about the version of the library at the beginning of the introduction.
28.3	2013/09/30	Add the macro `\makefirstname`
28.4	2013/10/09	Remove references to `fancyhdr`

Contents

List of Figures

List of Tables

1

Introduction

This documentation is written for and compiled by the
version 20131009
of tex-upmethodology.

This set of package permits to write documents according to Unified Process Methodology. It was initially wittren by Stéphane Galland from the laboratory "Systèmes et Transports"[1] and is distributed by the *Arakhnê*.org website. The provided packages and classes may also be used for other types of documents (reports, thesis...) Since 2012, it is used to support the layout and the style for the PhD thesis of the Doctoral School SPIM[2].

Packages are:

- `upmethodology-version.sty`: permits to set the version and the status of the document. It also permits to manage the document history;
- `upmethodology-fmt.sty`: provides some usefull functions to format the UP documents;
- `upmethodology-document.sty`: provides functions to manage the project, the subproject and the status of the document;
- `upmethodology-frontpage.sty`: formats and provides a front page for the document;
- `upmethodology-backpage.sty`: formats and provides a back page for the document;
- `upmethodology-task.sty`: is the LaTeX 2_ε package that provides macros to manage project's tasks.
- `upmethodology-document.cls`: is the LaTeX 2_ε class that provides the whole document specification. It is based on `book` and on the previous packages;
- `upmethodology-code.sty`: provides macros for source code formatting;
- `upmethodology-extension.sty`: provides macros for extension mechanism.

[1] Laboratory *Systèmes et Transport* (IRTES-SET), Institut de Recherche sur le Transport, l'Énergie et la Société (ITRTES), Université de Technologie de Belfort-Montbéliard (UTBM), France, http://set.utbm.fr/

[2] Doctoral School on the Sciences for enginers, and microtechnics, http://ed-spim.univ-fcomte.fr/

I

General User Documentation

2

Download and Installation

This chapter describes were to download `tex-upmethodology` and how to install it.

2.1/ Download

`tex-upmethodology` is available on the *Arakhné*.org website: http://www.arakhne.org/tex-upmethodology/.

Different types of installation are available: manual installation, Debian packages.

2.2/ Manual System-wide Installation

To make `tex-upmethodology` available to all users, copy the content of the `tex-upmethodology` archive inside one of your system texmf directory, usually one of:

- `/usr/share/texmf-texlive/tex/latex/upmethodology`,
- `/usr/share/texmf/tex/latex/upmethodology`.

The second is to rebuild the LaTeX databases by invoking on a console (Unix syntax us used):

```
$> sudo mktexlsr
$> sudo update-updmap --quiet
```

`sudo` is a standard Linux tool which permits to authorized users to temporarely obtain the administration rights.

2.3/ Manual User-wide Installation

To make `tex-upmethodology` available to one user, copy the content of the `tex-upmethodology` archive inside the `$HOME/texmf` directory.

It is not required to rebuild the system-wide LaTeX databases because the user's texmf are dynamically parsed by the LaTeX distributions.

2.4/ Debian Package Installation

Debian packages are available on *Arakhné*.org website: http://www.arakhne.org/ubuntu.html. Please follow the given rules.

2.5/ Package Dependencies

This section contains the list of all the package dependencies for the `upmethodology` packages.

2.5.1/ upmethdology-backpage.sty

`upmethodology-backpage` package depends on:

- `upmethodology-p-common`
- `upmethodology-extension`

2.5.2/ upmethdology-code.sty

`upmethodology-code` package depends on:

- `upmethodology-p-common`

2.5.3/ upmethdology-document.cls

`upmethodology-document` class depends on:

- `upmethodology-p-common`
- `a4wide`
- `upmethodology-document`
- `upmethodology-extension`
- `upmethodology-frontpage`
- `upmethodology-backpage`
- `upmethodology-task`
- `upmethodology-code`
- `upmethodology-spec`
- `url`
- `hyperref`

2.5.4/ upmethdology-document.sty

`upmethodology-document` package depends on:

- `upmethodology-p-common`
- `babel`
- `vmargin`
- `upmethodology-extension`
- `upmethodology-fmt`
- `upmethodology-version`

2.5.5/ UPMETHDOLOGY-EXTENSION.STY

`upmethodology-extension` package depends on:

- `upmethodology-p-common`

2.5.6/ UPMETHDOLOGY-FMT.STY

`upmethodology-fmt` package depends on:

- `upmethodology-p-common`
- `graphicx`
- `subfigure`
- `tabularx`
- `multicol`
- `colortbl`
- `picinpar`
- `amsmath`
- `amsthm`
- `thmtools`
- `pifont`
- `setspace`
- `txfonts`
- `xkeyval`
- `hyphenat`
- `mathbb`

2.5.7/ UPMETHDOLOGY-FRONTPAGE.STY

`upmethodology-frontpage` package depends on:

- `upmethodology-p-common`
- `upmethodology-extension`
- `upmethodology-document`

2.5.8/ UPMETHDOLOGY-P-COMMON.STY

`upmethodology-p-common` package depends on:

- `ifthen`
- `xspace`
- `xcolor`

2.5.9/ UPMETHDOLOGY-SPEC.STY

`upmethodology-spec` package depends on:

- `upmethodology-p-common`
- `ulem`
- `upmethodology-fmt`
- `upmethodology-code`

2.5.10/ UPMETHDOLOGY-TASK.STY

`upmethodology-task` package depends on:

- `upmethodology-p-common`
- `upmethodology-version`

2.5.11/ UPMETHDOLOGY-VERSION.STY

`upmethodology-version` package depends on:

- `upmethodology-p-common`
- `upmethodology-fmt`

II

Package Documentation

3

Class upmethodology-document

Version: 2013/10/09

The LaTeX class upmethodology-document provides the basic configuration for a document. According to an option, this class is able to extend the standard book, report or article LaTeX classes. It also include several of the other upmethdology packages.

3.1/ Types of documents

upmethodology-document supports three particular options, which permit to set the type of document:

- book: A book-specification is a two-sided document composed of parts and chapters, and with a copyright page and document information page. This option indicates to upmethodology-document to load the LaTeX standard book class. In addition the \part and \chapter macros are supported, and the following macros are automatically expanded: \makefrontcover, \upmpublicationpage, \upmdocumentsummary, \makebackcover. This behaviour may be overridden by the other class options.

- report: A report-specification is a one-sided document composed of chapters (no part), and with a document information page. This option indicates to upmethodology-document to load the LaTeX standard report class. In addition the \part macro is ignored[1] and \chapter macro is supported, and the following macros are automatically expanded: \makefrontcover, \upmdocumentsummary, \makebackcover. This behaviour may be overridden by the other class options.

- article: An article-specification is a one-sided document composed of sections (no part nor chapter). This option indicates to upmethodology-document to load the LaTeX standard article class. In addition the \part and \chapter macros are ignored[1], and the following macros are automatically expanded: \makefrontcover, \makebackcover. This behaviour may be overridden by the other class options.

3.2/ Class options

Table 3.1 contains the options supported by upmethodology-document. Any option not explicitely supported by the class is directly pass to the underlying standard LaTeX class (book, report or article according to the type of document, see 3.1).

[1] The macro is redefined to print a warning message when used, no error message is generated.

Options of upmethodology-document class	
Option	*Explanation*
book	see section 3.1.
report	see section 3.1.
article	see section 3.1.
oneside	the document is generated assuming that each page will be printed on its recto side. This option overrides any previous occurrence of `twoside` option.
twoside	the document is generated assuming that each page will be printed on both recto and verso sides. This option overrides any previous occurrence of `oneside` option.
francais	same as `french`.
french	the document is written in French. `upmethodology` packages use the French translations for the generated texts. This option overrides any previous occurrence of `english` option.
english	the document is written in English. `upmethodology` packages use the English translations for the generated texts. This option overrides any previous occurrence of `french` option.
documentinfo	invoke `\upmdocumentsummary`, `\upmdocumentauthors`, `\upmdocumentvalidators`, `\upmdocumentinformedpeople`, and `\upmhistory` macros at the begining of the document. This option overrides any previous occurrence of `nodocumentinfo` option.
nodocumentinfo	do not invoke `\upmdocumentsummary`, `\upmdocumentauthors`, `\upmdocumentvalidators`, `\upmdocumentinformedpeople`, nor `\upmhistory` macros at the begining of the document. This option overrides any previous occurrence of `documentinfo` option.
pubpage	invoke `\upmpublicationpage` macro at the begining of the document. This option overrides any previous occurrence of `nopubpage` option.
nopubpage	do not invoke `\upmpublicationpage` macro at the begining of the document. This option overrides any previous occurrence of `pubpage` option.

Table 3.1: Options of `upmethodology-document` class

3.3/ Additional Features

`upmethodology-document` provides a constant behaviour for all types of document:

- `hyperref` is loaded and set with the document informations;
- `\setpdfcolor` is redefined and linked to `hyperref`;

4

Package upmethodology-version

Version: 2013/08/26

The package upmethodology-version permits to set the version and the status of the document. It also provides functions to manage the document history;

4.1/ Constants for the Document Status

Some LaTeX 2_ε variables provides strings that describe the status of the document. They can be used in functions such as \updateversion.

- \upmrestricted: the document is under a restricted access, generally corresponding to the list of authors;
- \upmvalidable: authors indicates with this flag that the document could be sent to validators;
- \upmvalidated: the document was validated, but not published;
- \upmpublic: the document published and accessible to all people;

4.1.1/ Information about the Document

The following functions permit to access to the informations about the document:

- \theupmversion: replies the last version number for the document;
- \upmdate{version}: replies the updating date of the document corresponding to the given version number;
- \upmdescription{version}: replies the updating comment of the document corresponding to the given version number;
- \upmstatus{version}: replies the status of the document corresponding to the given version number.
- \theupmdate: replies the last updating date for the document. It is equivalent to \upmdate{\theupmversion};
- \theupmlastmodif: replies the last updating comment for the document. It is equivalent to \upmdescription{\theupmversion};
- \theupmstatus: replies the last status for the document. It is equivalent to \upmstatus{\theupmversion};

4.2/ Register Revisions

The package `upmethodology-version` permits to register revisions for building an history. The available functions are:

- `\updateversion{version}{date}{description}{status}`: registers a revision for the document. The revision indicates that the given version was produced at the given date. A small description of the changes and the resulting document's status must be also provided. The function `\updateversion` is a generalization of the following functions;
- `\initialversion[version]{date}{description}{status}`: registers the initial version of the document. If not given, the version is assumed to be `0.1`;
- `\incversion{date}{description}{status}`: regiters a revision corresponding to the next major version. For example, if the version number was `2.67` before `\incversion`, this function add the version `3.67` with the given informations (incrementation of the major part of the version number);
- `\incsubversion{date}{description}{status}`: regiters a revision corresponding to the next minor version. For example, if the version number was `2.67` before `\incsubversion`, this function add the version `2.68` with the given informations (incrementation of the minor part of the version number);

4.3/ Formatted List of Versions

To obtain a formatted list of versions, you could use the macro `\upmhistory[width]` which produces:

Version History		
Version	*Date*	*Updates*
18.0	2012/12/21	\upmuse → \UseExtension
19.0	2013/04/28	Replace the package inclusion of `color.sty` by `xcolor.sty`
20.0	2013/05/17	Add a section on the default configuration of the package `graphicx`
21.0	2013/05/30	Add the macros `\includegraphicswtex` and `\DeclareGraphicsExtensionsWtex`
21.1	2013/05/31	Fixing some grammar and syntax problems.
21.2	2013/07/06	Add the macro `\ifdocumentauthor`.
22.0	2013/08/15	Add the macros: `\textup`, `\textsubscript`, `\textdown`, `\textsubscript`, `\Emph`. Add the environments `enumdescription`, `enumdescriptionx`.
23.0	2013/08/17	Update the second parameter of `\mfigurewtex` to have the same semantic as for `\mfigure`.
24.0	2013/08/26	Major bug fixes in the environments `mtabular` and `mtable`. Add the macro `\tabulartitlespec`. Add the macro `\makenamespacing`.
25.0	2013/08/27	Add the environment `framedcolorminipage`. Add the `definition` environment.
26.0	2013/09/02	Change the specification of the optional parameter of `mtable`. Add the macro `\makelastname`
27.0	2013/09/05	Add the macro `\declareupmtheorem`
28.0	2013/09/19	Add the mathematic symbols
28.1	2013/09/26	Add the missed table placement flag `p`
28.2	2013/09/29	Add a notification text about the version of the library at the beginning of the introduction.
28.3	2013/09/30	Add the macro `\makefirstname`
28.4	2013/10/09	Remove references to `fancyhdr`

4.4/ Localization

The following macros defines some localized strings used by `upmethodology-version`:

- `\upm@lang@date`: Date;
- `\upm@lang@updates`: Updates;
- `\upm@lang@version`: Version;
- `\upm@lang@version@history`: Version History;

5

Package upmethodology-fmt

Version: 2013/10/01

The package upmethodology-fmt provides some usefull facilities to format a document.

5.1/ Default Configuration for the Package graphicx

The package graphicx is included, and the following configuration is applied:

- **Image extensions:** By default, the supported image extensions ar, in the preference order: pdf, png, jpg, jpeg, tiff, gif. Note that, the tiff picture format is not always supported by the TEX tools.
  To redefine these extensions, you must invoke:
  \DeclareGraphicsExtensions{extensions}
  where extensions must be replaced by a list of extensions separated by comas.
  Example: \DeclareGraphicsExtensions{.pdf,.png,.eps}

- **Image search path:** By default, the images are search inside the path "./". To redefine the search paths, you must invoke: \graphicspath{{path1},{path2},{path3}...}
  where path1, path2, path2, etc. must be replaced by the names of the directories in which the images are located. The paths in the list are separated by comas. *Do not forget to write a slash or a backslash character (depending on the path naming conventions for your operating system) at the end of each path.*
  Example: \graphicspath{{./imgs/},{./imgs/auto/}}

5.2/ Figures

It may be verbose to put LATEX code to include a figure inside your document. To simplify your life, you could include a figure with the following macros:
\mfigure[position]{include_graphics_options}{filename}{caption}{label}
\mfigure*[position]{include_graphics_options}{filename}{caption}{label}

These two macros permits to include an image in your document. The parameters are:

- position: is the desired position of the figure (see \begin{figure}[position]). It could be t (top of the page), b (bottom of the page), h (at the macro location if possible) or H (at macro location);
- include_graphics_options: are the options passed to \includegraphics;

- `filename`: is the filename passed to `\includegraphics`;
- `caption`: is the caption of the figure (see \caption{caption});
- `label`: is the label used to reference the figure (see \label{fig:label}).

The difference between `\mfigure` and `\mfigure*` is the same as the difference between `\begin{figure}` and `\begin{figure*}`: the star-version fits to the entire paper width event if the document has two or more columns.

Because the two macros above register a label with string starting with `fig:`, we propose the following function to easily access to the figure's references:

- `\figref{label}`: is equivalent to `\ref{fig:label}`;
- `\figpageref{label}`: is equivalent to `\pageref{fig:label}`.

The figure 5.1 page 30 is obtained with the macro: `\mfigure[ht]{width=.4\linewidth}{slogo}{Example of figure inclusion with \texttt{\textbackslash mfigure}}{example:mfigure}`.
The reference and page reference are obtained with `\figref{example:mfigure}` and `\figpageref{example:mfigure}`.

Figure 5.1: Example of figure inclusion with `\mfigure`

5.3/ SUB-FIGURES

In some case, it is usefull to put several images inside the same floating figure, but without loosing the possibility to reference each of the subfigures. This feature was proposed by the package `subfigure`. The following environments provides helper functions for `subfigure`:

```
\begin{mfigures}[position]{caption}{label}
...
\end{mfigures}
\begin{mfigures*}[position]{caption}{label}
...
\end{mfigures*}
```

These two macros permits to include an image in your document.The parameters are:

- `position`: is the desired position of the figure (see \begin{figure}[position]). It could be `t` (top of the page), `b` (bottom of the page), `h` (at the macro location if possible) or `H` (at macro location);
- `caption`: is the caption of the figure (see \caption{caption});
- `label`: is the label used to reference the figure (see \label{fig:label}).

Inside the environment `\mfigures[*]`, you could use the macro `\mfigure` to properly include a subfigure (the first optional parameter is ignored), or you could use the macro `\msubfigure{options}{file}{caption}`.

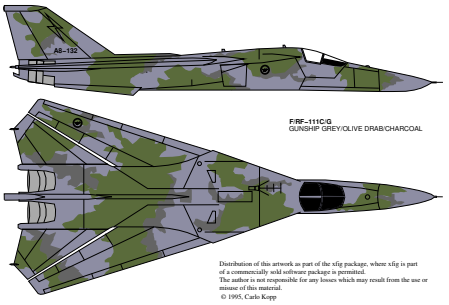
The figure 5.2 page 31 is obtained with the environment:

```
\begin{mfigures}{Example of subfigures with \texttt{mfigures}}{example:msubfigure}
\mfigure{width=.4\linewidth}{img1}{First subfigure}{example:firstsubfigure}
\hfill
\msubfigure{width=.4\linewidth}{img2}{Second subfigure}
\end{mfigures}
```

The reference and page reference are obtained with `\figref{example:msubfigure}` and `\figpageref{example:msubfigure}`.

(a) First subfigure

Example of a figure combined



(b) Second subfigure

Figure 5.2: Example of subfigures with `mfigures`

The references to the subfigures could be obtained in two way:

- using the label given as the last parameter of `\mfigure`, eg. the label `example:firstsubfigure` corresponds to 5.2(a);
- using the label of the enclosing figure to which the index of the subfigure could be appended (in its roman representation and prefixed by the character "`:`"), eg. the label `example:msubfigure:b` corresponds to 5.2(b);

5.4/ FIGURES WITH EMBEDDED TEX MACROS

In several case it is usefull to include TEX macros inside a figure. It is possible to combine figures and TEX macros. Several figure editors provide exporting features to obtain combined figures: `xfig`, `inkscape`, `GNU Plot`, etc. Basically, these tools create two files per source figure:

- the figure in PDF or Postscript format (filename extensions, `.pdf` or `.ps`); and
- a TEX file that contains the macros to put over the figure, and that is including the generated figure. Its filename extension depends on the type of the figure: `.pdftex_t` or `.pdf_tex` for PDF, and `.pstex_t` or `.ps_tex` for Postscript.

To include this combined figure in your document, you simply need to include the generated TEX file (see below for details).

5.4.1/ INCLUDE A COMBINED PICTURE/TEX FIGURE

To include a figure with TEX macros inside, you must have:

1. a Postcript figure (`.eps`), and a TEX file `.pstex_t` related to the Postscript figure; or
2. a PDF figure (`.pdf`), and a TEX file `.pdftex_t` related to the PDF figure.

With the `upmethodology-fmt` package, the inclusion of the figure with embedded TEX macros is similar to the inclusion of figures with `\includegraphics`. You must type the following macro:
`\includegraphicswtex[options]{filename}`
where `options` must be one or more of:

- `width=xxx`: specification of the width of the figure (`xxx` must be replaced by the length);
- `height=xxx`: specification of the height of the figure (`xxx` must be replaced by the length);

If the `filename` given to the macro `\includegraphicswtex` does not specify a filename extension, the macro tries to add the extensions `.pdftex_t`, `.pstex_t`, `.pdf_tex`, or `.ps_tex`, by default. If you want to specify other file extensions, you must use the macro: `\DeclareGraphicsExtensionsWtex{extensions}`
where the `extensions` is a list of file extensions (including the point character), separated by coma characters.
Example: `\DeclareGraphicsExtensionsWtex{.pdftex,.pstex}`

If the `filename` does not correspond to a file on the disk, the macro `\includegraphicswtex` tries to find the file in the directories specified in `\graphicspath` (declared in the package `graphicx` for example).
Example: `\graphicspath{{./imgs/},{./imgs/additional/}}`
Note that each of the given directories must be finished by the separation character of your operating system: / on Unix, \ on Windows. You must always use the Unix standard because it is assumed by a lot of TEX compilers, even on Windows platforms.

Figure 5.3 gives an example of a floating figure combined with TEX macros, which is using the macro `\includegraphicswtex`.

5.4.2/ Floating figure with embedded TEX macros

To put a floating figure with TEX macro inside, you may use one of the macros:
`\mfigurewtex[position]{include_graphics_options}{filename}{caption}{label}`
`\mfigurewtex*[position]{include_graphics_options}{filename}{caption}{label}`

The parameters are:

- `position`: is the desired position of the figure (see \beginfigure[position]). It could be `t` (top of the page), `b` (bottom of the page), `h` (at the macro location if possible) or `H` (at macro location);
- `include_graphics_options`: are the options to pass to `\includegraphicswtex`. For ascendent compatibility, if you pass a length without a key, e.g. `{.8\linewidth}`, the length is assumed to be the width of the figure;
- `filename`: is the name of the file of the figure (see `\includegraphicswtex` for details);
- `caption`: is the caption of the figure (see \caption{caption});
- `label`: is the label used to reference the figure (see \label{fig:label}).

The difference between `\mfigurewtex` and `\mfigurewtex*` is the same as the difference between `\begin{figure}` and `\begin{figure*}`: the star-version fits to the entire paper width event if the document has two or more columns.

Because the two macros above register a label with string starting with `fig:`, the macros `\figref` and `\figpageref` could be used.

Figure 5.3 gives an example of a floating figure combined with TEX macros. Note that:

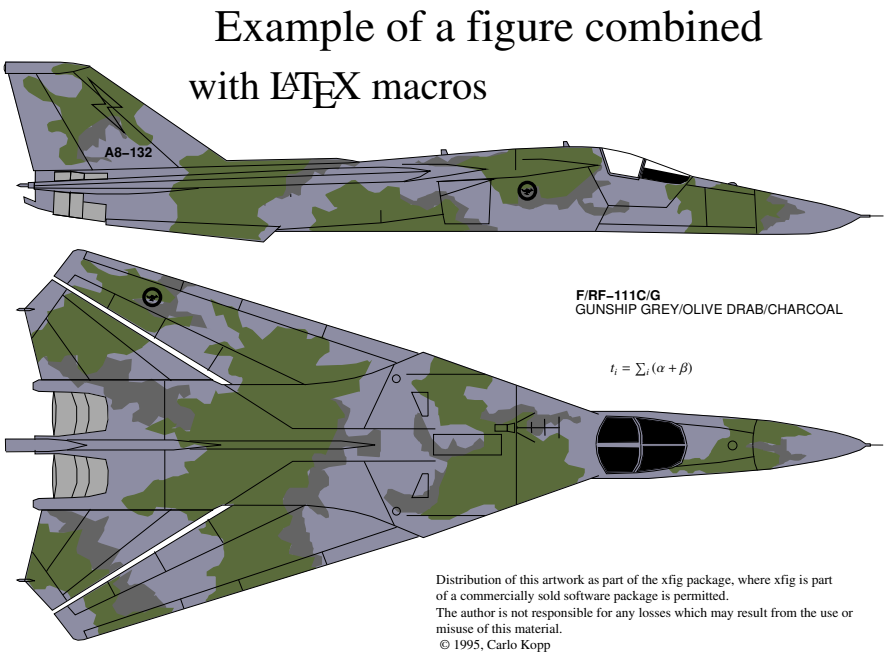


Figure 5.3: Example of a figure combined with TeX macros

- the title of the figure contains the macro `\LaTeX`, which produces: LaTeX;
- a small equation, written in TeX, is put between the two planes;

5.4.3/ Helpers for embedded TeX

To help you to put TeX macros in a figure, and to define its real test inside the LaTeX document, several functions are provided:

- `\figmath{id}{expr}` will associate to the given identifier the given mathematical expression,
- `\figtext{id}{expr}` will associate to the given identifier the given text expression;

These expressions, defined with the two previous functions, may be referenced in the figure by a TeX macro with a name similar to `\FIG`δ, where δ must be replaced by an identifier of your choice and used as parameter of one of the two previous functions (example: `\FIGmyid`).

Figure 5.3 gives an example where the equation is written as: `\FIGexampleofexpression` in the figure, and it is replaced by the real equation with:
`\figmath{exampleofexpression}{t_i = \sum_i \left(\alpha + \beta\right)}`

5.5/ Tabulars

You could include a tabular inside your document with the following environment:
`\begin{mtabular}[width]{ncolumns}{columns}...\end{mtabular}`

This tabular is an extension of the `tabularx` environment which provides dynamic columns with the specifier `X`. The parameters are:

- `width`: is the desired width of the tabular;

- `ncolumns`: is the count of columns in the tabular. It must be consistent with the column description;
- `columns`: is the description of the columns according to the `tabular` and `tabularx` packages.

You must note put any text nor TEX macro before the first use of `\tabulartitle` or `\tabularheader`. Otherwise, you will obtain an TEX error.

The `mtabular` environment provides:

- `\tabulartitle{title}`
  This macro permits to define the title of the tabular. It uses the colors `backtableheader` and `fronttableheader` for the background and the foreground respectively. The title has a single line at the top, and a single line below;
- `\tabulartitleinside{title}`
  This macro permits to define the title of the tabular. It uses the colors `backtableheader` and `fronttableheader` for the background and the foreground respectively. The title has two lines at the top, and a single line below;
- `\tabularheader{`$header_1$`}...{`$header_n$`}`
  This macro permits to define the titles of the columns. It uses the colors `backtableheader` and `fronttableheader` for the background and the foreground respectively. Because the count of columns was given to the environment this function takes the same count of parameters as the count of columns. This macro adds a line after the header, *BUT NOT BEFORE*.

  Because `\tabularheader` is adding a `\hline` at the end of its expansion. You must put a `\tabularheader` just after `\tabularheader`. Otherwise you may obtain a TEX error.

- `\tabularrowheader{title}`
  This macro is designed to be used in the first cell of a row. It is rendering the cell as a row's header.
- `\tabulartitlespec{column_spec}`
  This macro defines the specification of the column used to render the title of the table. The default value of the column specification is `|c|`.

The following example of table is obtained by:

```
\begin{mtabular}[\linewidth]{4}{lXrX}
\tabulartitle{Example of \texttt{mtabular}}
\tabularheader{Col1}{Col2}{Col3}{Col4}
a & b & c & d \\
\hline
e & f & g & h \\
\tabulartitleinside{Example of second title in the table}
\hline
\tabularrowheader{i} & j & k & l \\
\tabularheader{Col1-2}{Col2-2}{Col3-2}{Col4-2}
m & n & o & p \\
\end{mtabular}
```

Example of `mtabular`			
Col1	*Col2*	*Col3*	*Col4*
a	b	c	d
e	f	g	h
Example of second title in the table			
i	j	k	l
Col1-2	*Col2-2*	*Col3-2*	*Col4-2*
m	n	o	p

5.6/ TABLES

You could include a table inside your document with the following environment:
`\begin{mtable}[options]{width}{ncolumns}{columns}{caption}{label}...\end{mtable}`

This environment is based on the `mtabular` environment. The parameters are:

- `options`: are the options to pass to the `mtable` environment:
  - a table placement composed of one or more of the following characters. The order in which the placement options are specified does not make any difference, as the placement options are always attempted in the order `h-t-b-p`. Thus `[hb]` and `[bh]` are both attempted as `h-b`. The more float placement options are given to LaTeX, the better it handles float placement. Consequently, and because we want a simple TeX code in the background, all the permutations are not supported by the `mtable` environment. We recommend to put placement letters in the order they appear in the following list:
    - `h`: Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot),
    - `t`: Position at the top of the page,
    - `b`: Position at the bottom of the page,
    - `p`: Put on a special page for floats only,
    - `H`: Places the float at precisely the location in the LaTeX code. Requires the `float` package. This is somewhat equivalent to `h!`.;
    - `!`: Override internal parameters LaTeX uses for determining "good" float positions,

    If you specify more than one table placement in the options, the last one is used.
  - `size=<macro>`: specify the size of the text in the table (by default, `\normalsize`);
- `width`: is the desired width of the table (ie., the tabular inside the table);
- `ncolumns`: is the count of columns in the table (ie., the tabular inside the table). It must be consistent with the column description;
- `columns`: is the description of the columns according to the `tabular` and `tabularx` packages;
- `caption`: is the caption of the table;
- `label`: is the label referencing the table.

Because the `mtable` environment registers a label with a string starting with `tab:`, the following functions are proposed to easily access to the table's references:

- `\tabref{label}`: is equivalent to `\ref{tab:label}`;
- `\tabpageref{label}`: is equivalent to `\pageref{tab:label}`.

Example of `mtable`			
Col1	*Col2*	*Col3*	*Col4*
a b		c d	
e f		g h	

Table 5.1: Example of `mtable`

The table 5.1 page 36 is an illustration of the following LaTeX code:

```
\begin{mtable}{\linewidth}{4}{lXrX}{Example of \textttmtable}{example:mtable}
\captionastitle
\tabularheader{Col1}{Col2}{Col3}{Col4}
a & b & c & d \\
\hline
e & f & g & h \\
\end{mtable}
```

The macro `\captionastitle` is equivalent to a call to the macro `\tabulartitle` with the caption in parameter.

5.7/ Enumerations

The package `upmethodology-fmt` provides a set of macros dedicated to enumeration lists.

5.7.1/ Enumeration Counters

Sometimes it is usefull to start an enumeration list from a specifical given number. This package provides several macros for saving and restoring the counter use by the enumeration lists.

Only once counter could be saved at a given time. It means that you cannot save the counters for an enumeration and for an enclosing enumeration at the same time.

- `\savecounter{name}`
  save the counter identifier by the given name;
- `\restorecounter{name}`
  put the previously saved value into the given counter;
- `\setenumcounter{value}`
  force the value of the enumeration counter;
- `\getenumcounter`
  replies the value of the enumeration counter;
- `\saveenumcounter`
  save the enumeration counter;
- `\restoreenumcounter`
  force the enumeration to use the saved counter's value;

5.7.2/ Inline Enumeration

In several document, an enumeration of things is written inside a paragraph instead of inside a list of points.

Exemple: The following LaTeX code produces the result below:

```
This is a text: \begin{inlineenumeration}
\item first thing;
\item second thing;
\item etc.
\end{inlineenumeration} This is the text after.
```

This is a text: (i) first thing; (ii) second thing; (iii) etc. This is the text after.

5.8/ Descriptions in conjonction with enumeration

It may be helpful to put a list of descriptions in conjonction with an enumeration. In other words, the following environment provides a mix between the standards LaTeX environments `description` and `enumerate`.

5.8.1/ Environment `enumdescription`

The environment `enumdescription` is:

```
\begin{enumdescription}[type]
\item[desc] text
\end{enumdescription}
```

where the `type` is the type of the enumeration. It may be one of:

- "`i`": for an enumeration with roman numbers (this is the default),
- "`1`": for an enumeration with arabic numbers,
- "`a`": for an enumeration with letters.

The text put in place of `desc` represents the text which may be emphazed in the description item.

To change the rendering of the labels, you must redefined the macro as:

```
\renewcommand{\enumdescriptionlabel}[1]{ ... #1 ... }
```

Example 1: The following LaTeX code, using roman numbers, produces the enumerated description just below:

```
\begin{enumdescription}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}
```

i - first thing: this is a text for the first thing;

ii - second thing: this is a text for the second thing;

iii - more: etc.

Example 2: The following LaTeX code, using numeric numbers, produces the enumerated description just below:

```
\begin{enumdescription}[1]
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}
```

1 - first thing: this is a text for the first thing;

2 - second thing: this is a text for the second thing;

3 - more: etc.

Example 3: The following LaTeX code, using letter numbers, produces the enumerated description just below:

```
\begin{enumdescription}[a]
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}
```

a - first thing: this is a text for the first thing;

b - second thing: this is a text for the second thing;

c - more: etc.

5.8.2/ Environment `enumdescriptionx`

The environment `enumdescriptionx` extends the environment `enumdescription` by enabling a finer configuration with more parameters.

The environment `enumdescriptionx` is:

```
\begin{enumdescriptionx}[type]{counter\_prefix}{counter\_postfix}
\item[desc] text
\end{enumdescriptionx}
```

where the `type` is the type of the enumeration. It may be one of:

- "`i`": for an enumeration with roman numbers (this is the default),
- "1": for an enumeration with arabic numbers,
- "a": for an enumeration with letters.

The text put in place of `desc` represents the text which may be emphazed in the description item. The text `counter_prefix` is put before all the counter values in the enumeration. The text `counter_postfix` is put after all the counter values in the enumeration.

To change the rendering of the labels, you must redefined the macro as:

```
\renewcommand{\enumdescriptionlabel}[1]{ ... #1 ... }
```

Example: The following LaTeX code, using letter numbers, produces the enumerated description just below:

```
\begin{enumdescriptionx}[a]{$\langle$}{$\rangle$}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescriptionx}
```

⟨**a**⟩ **- first thing**: this is a text for the first thing;

⟨**b**⟩ **- second thing**: this is a text for the second thing;

⟨**c**⟩ **- more**: etc.

5.9/ Footnotes

The package `upmethodology-fmt` provides a set of macros allowing to save the reference number of a footnote and to recall this reference many time as required.

- `\savefootnote{footnote text}{footnote id}`
  put a footnote and mark it with the corresponding label.
  Example: `\savefootnote{This is an example of a recallable footnote}{footrecalla}`[1];

- `\savefootnote*{footnote text}{footnote id}`
  mark a footnote with the corresponding label but do not put in the current page.
  Example 1: `\savefootnote*{This is a second example of a recallable footnote}{footrecallb}`;
  Example 2: `\savefootnote*{This is a third example of a recallable footnote}{footrecallc}`.

- `\reffootnote{footnote id}`
  recall the footnote reference without page number.
  Example 1: `\reffootnote{footrecalla}`[1] = *B*;
  example 2: `\reffootnote{footrecallb}`[2] = *A*;
  example 4: `\reffootnote{footrecalld}`[??] =?.

- `\reffootnote*{footnote id}`
  recall the footnote reference with the page number if different of the current page.
  Example 1: `\reffootnote*{footrecalla}`[1(39)];
  example 2: `\reffootnote*{footrecallb}`[2(39)];
  example 3: `\reffootnote*{footrecallc}`[3];
  example 4: `\reffootnote*{footrecalld}`[??(??)].

5.10/ UML diagrams on the side of paragraphs

The package `upmethodology-fmt` provides an environment which permits to put an UML diagram (or any other picture) on the side of a paragraph.

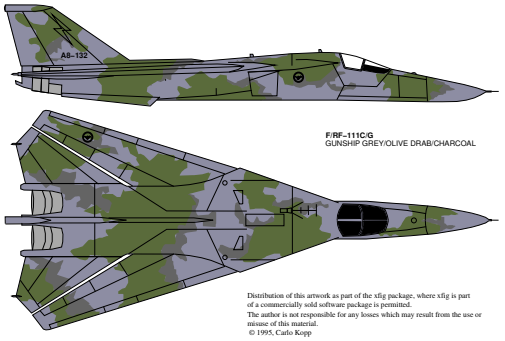
[1] This is an example of a recallable footnote
[2] This is a second example of a recallable footnote
[3] This is a third example of a recallable footnote

- `\begin{umlinpar}[width]{picture_path}`
  `text`
  `\end{umlinpat}`
  put the specified picture on the side of the given text. The optional parameter `width` corresponds to the desired width ofthe picture. By default it is `.5\linewidth`.

This paragraph is an typical example of the usage of the environment `umlinpar`. To obtain it, the following LaTeX code was typed:

```
\begin{umlinpar}{smalllogo}
This paragraph is an typical example
of the usage of the environment
\texttt{umlinpar}.
\end{umlinpar}
```

Example of a figure combined



5.11/ Date formatting

Because the concept of date was important and unfortunately localized, this package provides a set of functions to define and extract information from dates (the supported date formats are described in table 5.2):

- `\makedate{day}{month}{year}`
  permits to create the text corresponding to the given date according to the current localized date format.
- `\extractyear{formatted_date}`
  extract the year field from a date respecting the localized date format.
- `\extractmonth{formatted_date}`
  extract the month field from a date respecting the localized date format.
- `\extractday{formatted_date}`
  extract the day field from a date respecting the localized date format.

`yyyy/mm/dd`	default format
`dd/mm/yyyy`	french format

Table 5.2: List of supported date formats

5.12/ Text formatting

The package `upmethodology-fmt` provides a set of macros to format the text.

- `\textsup{text}`
  put a text as exponent in text mode instead of the basic LaTeXexponent in math mode. In opposite to the standard LaTeX macro `\textsuperscript`, this macro adds an extra space after the macro when needed.
  Example: `\textsup{this is an exponent}`this is an exponent this is the following text;
- `\textup{text}`
  same as `\textsup`.

- \textsub{text}
  put a text as indice in text mode instead of the basic LaTeXindice in math mode. In opposite to \textsubscript, this macro adds an extra space after the macro when needed. In opposite to \textdown, the size of the text is not changed in the text down.
  Example: \textsub{this is an indice}$_{\text{this is an indice}}$ this is the following text;

- \textdown{text}
  put a text as indice in text mode instead of the basic LaTeXindice in math mode. In opposite to \textsubscript, this macro adds an extra space after the macro when needed. In opposite to \textsub, the size of the text is changed in the text down.
  Example: \textdown{this is an indice}$_{\text{this is an indice}}$ this is the following text;

- \textsubscript{text}
  put a text as indice in text mode instead of the basic LaTeXindice in math mode. As for the standard LaTeX macro \textsuperscript, this macro does not add an extra space after the macro.
  Example: \textsubscript{this is an indice}$_{\text{this is an indice}}$this is the following text;

- \Emph{text}
  put a *very important* text. This macro is similar to the standard LaTeX macro \emph. The difference is: \emph is for "important things"; and \Emph is for "very important things".
  Example: This text is \emph{important}, but this one is \Emph{very important}
  gives: This text is *important*, but this one is **very important**;

- \makename[von]{first name}{last name}
  format the specified people name components according to the document standards. By default, the format first von last is used.
  Example: \makename[von]{Ludwig Otto Frederik Wilhelm}{Wittelsbach},
  "Ludwig Otto Frederik Wilhelm von Wittelsbach";

- \upmmakename[von]{first name}{last name}{separator}
  format the specified people name components according to the document standards. By default, the format first von last is used.
  Example: \upmmakename[von]{Ludwig Otto Frederik Wilhelm}{Wittelsbach}{/},
  "Ludwig Otto Frederik Wilhelm/von/Wittelsbach";

- \makenamespacing{name}
  format the specified name to be sure that the spaces after the points of the initials are demi-spaces.
  Example: \makenamespacing{S.G.}Galland,
  "S. G. Galland";

- \makelastname{name}
  format the specified last/family name.
  Example: \makelastname{Galland},
  "Galland";

- \makefirstname{name}
  format the specified first name.
  Example: \makefirstname{Stéphane},
  "Stéphane";

- \prname[von]{first name}{last name}
  \prname*[von]{first name}{last name}
  format the specified people name components according to the document standards for *Professor* title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: \prname{Pierre}{Martin}, "Pr. Pierre Martin";
  Example 2: \prname*{Pierre}{Martin}, "Pierre Martin, Pr.";

- \drname[von]{first name}{last name}
  \drname*[von]{first name}{last name}
  format the specified people name components according to the document standards for *Doctor* title.

By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
Example 1: `\drname{Pierre}{Martin}`, "Dr. Pierre Martin";
Example 2: `\drname*{Pierre}{Martin}`, "Pierre Martin, Dr.";

- `\phdname[von]{first name}{last name}`
  `\phdname*[von]{first name}{last name}`
  format the specified people name components according to the document standards for *Philosophiæ-Doctor* title. By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `\phdname{Pierre}{Martin}`, "Ph.D. Pierre Martin";
  Example 2: `\phdname*{Pierre}{Martin}`, "Pierre Martin, Ph.D.";

- `\scdname[von]{first name}{last name}`
  `\scdname*[von]{first name}{last name}`
  format the specified people name components according to the document standards for *ScientiæDoctor* title. By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `\scdname{Pierre}{Martin}`, "Sc.D. Pierre Martin";
  Example 2: `\scdname*{Pierre}{Martin}`, "Pierre Martin, Sc.D.";

- `\mdname[von]{first name}{last name}`
  `\mdname*[von]{first name}{last name}`
  format the specified people name components according to the document standards for *Medicinæ-Doctor* title. By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `\mdname{Pierre}{Martin}`, "M.D. Pierre Martin";
  Example 2: `\mdname*{Pierre}{Martin}`, "Pierre Martin, M.D.";

- `\pengname[von]{first name}{last name}`
  `\pengname*[von]{first name}{last name}`
  format the specified people name components according to the document standards for *Professional/Chartered Engineer* title. By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `\pengname{Pierre}{Martin}`, "CEng. Pierre Martin";
  Example 2: `\pengname*{Pierre}{Martin}`, "Pierre Martin, CEng.";

- `\iengname[von]{first name}{last name}`
  `\iengname*[von]{first name}{last name}`
  format the specified people name components according to the document standards for *Incorporated Engineer* title. By default, the format `first von last` is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `\iengname{Pierre}{Martin}`, "IEng. Pierre Martin";
  Example 2: `\iengname*{Pierre}{Martin}`, "Pierre Martin, IEng.".

5.13/ Symbols

5.13.1/ Symbols in Text Mode

The package `upmethodology-fmt` provides several symbols in text mode, and described inside the table 5.3.

5.13.2/ Symbols in Math Mode

The package `upmethodology-fmt` provides several symbols in math mode, and described inside the table 5.4.

\arakhneorg	Arakhnê.org
\copyright	©
\trademark	™
\regmark	®
\smalltrade	™
\smallreg	®
\smallcopy	©
\ust	st
\und	nd
\urd	rd
\uth	th

Table 5.3: List of symbols

Sets	
\R	$\mathbb{R}$
\N	$\mathbb{N}$
\Z	$\mathbb{Z}$
\Q	$\mathbb{Q}$
\C	$\mathbb{C}$
\powerset p	$\mathcal{P}p$
Operators	
\sgn expr	sgn *expr*

Table 5.4: List of symbols

5.14/ Bibliography

The package `upmethodology-fmt` provides a set of macros allowing to manage the bibliography. The default bibliography style is `abbr`.

- `\bibliographystyle{style}`
  set the bibliography style to use.
  Example: `\bibliographystyle{alpha}`;

- `\bibliography{file}`
  set the BibTeX file to use.
  Example: `\bibliography{mybib}`;

- `\bibsize{size}`
  set the font size used for the bibliography section.
  Example: `\bibsize{\Huge}`;

5.15/ Theorems and Mathematic Environments

The package `upmethodology-fmt` defines several environments and macros that are based on the `theorem` or the math API of LaTeX.

5.15.1/ Definition of a new theorem environment

If you want to create a new theorem environment based on the style provided by this package, you could invoke `\declareupmtheorem`:
`\declareupmtheorem[name of the style]{name}{label}{title of the list}`

This macro defines:

- the environment with the given name, and
- the macro `\listof⟨name⟩s`.

The `name of the style` is the name of the theorem style to be used. This style is defined by `\newtheoremstyle`. By default, it is `upmdefinition`. The `label` is the text to put in the theorem header. The `title of the list` is used by the macro `\listof⟨name⟩s` as the title of the chapter.

Some features provided by this package depend on the version of the package `thmtools`. We recommend to use and install the version 2012/05/04, or later.

The macro `\declareupmtheorem` can be used only inside the preamble of your document.

Example: The following code define the environment `mytheorem`:

```
\documentclass{upmethodology-document}
\declareupmtheorem{mytheorem}{My Theorem}{List of my Theorems}
\begin{document}
\begin{mytheorem}[Theorem of Everything]
This is the theorem of Evereything.
\end{mytheorem}
\end{document}
```

gives the result:

My Theorem 1: Theorem of Everything

This is the theorem of Evereything.

5.15.2/ DEFINITION

The package `upmethodology-fmt` defines the environment `definition`, which permits to put a definition in your document. This environment is based on the standard `theorem` environment. The `definition` takes one optional parameter: the name of the definition.

Example: The following LaTeX code:

```
\begin{definition}[Name of the definition]
Text of the definition.
\end{definition}
```

produces:

Definition 1: Name of the definition

Text of the definition.

Change the colors of the definition: You could change the colors of the `definition` environment by redefining the colors below with one of the macros `\definecolor` or `\colorlet`:

- `definitionbackground` is the color of the background of the definition;
- `definitionborder` is the color of the frame;
- `definitionheaderforeground` is the color of the text in the header of the definition;
- `definitiontextforeground` is the color of the text in the body of the definition.

Example of color redefinition: The following LaTeX code:

```
\definecolor{definitionheaderforeground}{rgb}{.3,.5,.8}
\colorlet{definitionbackground}{gray!20}
\colorlet{definitionborder}{red}
\begin{definition}[Name of the definition]
Text of the definition.
\end{definition}
```

produces:

Definition 2: Name of the definition

Text of the definition.

5.16/ Framed Boxes or Mini Pages

Standard LaTeX distribution provides the `minipage` environment. This environment permits to put a small piece of page inside your document. The package `upmethodology-fmt` provides two framed extensions of the original `minipage` environment: `framedminipage` and `framedcolorminipage`.

The prototypes of there two new environments are, respectively:

- `\begin{framedminipage}{width} ...\end{framedminipage}`
- `\begin{framedcolorminipage}{width}{border_color}{background_color} ...\end{framedcolorminipage}`

Example of `framedminipage` The following LaTeX code:

```
\begin{framedminipage}{.75\linewidth}
This is a text inside a framed minipage.
\end{framedminipage}
```

produces: This is a text inside a framed minipage.

Example of `framedcolorminipage` The following LaTeX code:

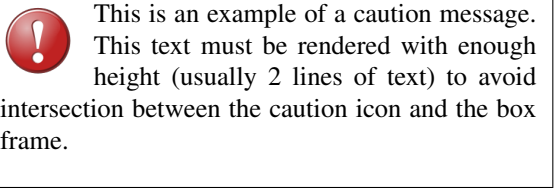
```
\begin{framedcolorminipage}{.75\linewidth}{red}{yellow}
This is a text inside a framed minipage with colors.
\end{framedcolorminipage}
```

produces: This is a text inside a framed minipage with colors.

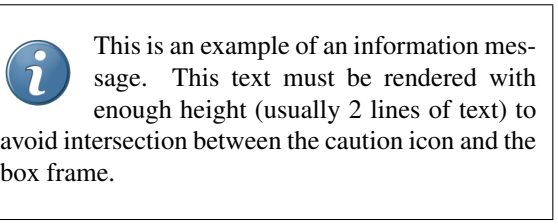
5.17/ Message Boxes

The package `upmethodology-fmt` provides a set of environment to put emphasis message boxes in the text. Three types of boxes are supported: caution, information, and question.

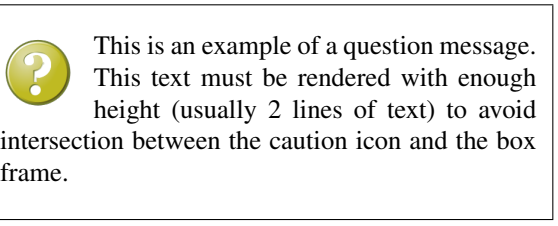
```
\begin{upmcaution}[width]
This is an example of a caution
message.  This text must be rendered
with enough height (usually 2 lines
of text) to avoid intersection between
the caution icon and the box frame.
\end{upmcaution}
```




```
\begin{upminfo}[width]
This is an example of an information
message.  This text must be rendered
with enough height (usually 2 lines
of text) to avoid intersection between
the caution icon and the box frame.
\end{upminfo}
```




```
\begin{upmquestion}[width]
This is an example of a question
message.  This text must be rendered
with enough height (usually 2 lines
of text) to avoid intersection between
the caution icon and the box frame.
\end{upmquestion}
```



5.18/ Additional Document Sectionning Macros

The package `upmethodology-fmt` provides several macros that permit to create special sections.

5.18.1/ Non-numbered Part in Table of Content

If you want to add a document part that has no part number but appearing inside the table of content, the classical LaTeX macros `\part` and `\part*` are inefficient. Indeed, `\part` is adding a numbered part inside the table of content, and `\part*` is adding an unnumbered part but not inside the table of content.

To add a unnumbered part inside the table of content, you could use one of the macros:
`\parttoc[toctitle]{title}`
`\parttoc*[toctitle]{title}`

The macros `\parttoc` and `\parttoc*` have the same effect except that `\parttoc*` aligns the part's title to the other numbered parts' titles; and `\parttoc` not.

5.18.2/ Non-numbered Chapter in Table of Content

If you want to add a document chapter that has no chapter number but appearing inside the table of content, the classical LaTeX macros `\chapter` and `\chapter*` are inefficient. Indeed, `\chapter` is adding a numbered chapter inside the table of content, and `\chapter*` is adding an unnumbered chapter but not inside the table of content.

To add a unnumbered chapter inside the table of content, you could use one of the macros:
`\chaptertoc[toctitle]{title}`
`\chaptertoc*[toctitle]{title}`

The macros `\chaptertoc` and `\chaptertoc*` have the same effect except that `\chaptertoc*` aligns the chapter's title to the other numbered chapters' titles; and `\chaptertoc` not.

5.18.3/ Non-numbered Section in Table of Content

If you want to add a document section that has no a section number but appearing inside the table of content, the classical LaTeX macros `\section` and `\section*` are inefficient. Indeed, `\section` add a numbered section inside the table of content, and `\section*` adds an unnumbered section but not inside the table of content.

To add a unnumbered section inside the table of content, you could use one of the macros:
`\sectiontoc[toctitle]{title}`
`\sectiontoc*[toctitle]{title}`

The macros `\sectiontoc` and `\sectiontoc*` have the same effect except that `\sectiontoc*` aligns the section's title to the other numbered sections' titles; and `\sectiontoc` not.

5.18.4/ Non-numbered Subsection in Table of Content

If you want to add a document subsection that has no subsection number but appearing inside the table of content, the classical LaTeX macros `\subsection` and `\subsection*` are inefficient. Indeed, `\subsection` is adding a numbered subsection inside the table of content, and `\subsection*` is adding an unnumbered subsection but not inside the table of content.

To add a unnumbered subsection inside the table of content, you could use one of the macros:
`\subsectiontoc[toctitle]{title}`
`\subsectiontoc*[toctitle]{title}`

The macros `\subsectiontoc` and `\subsectiontoc*` have the same effect except that `\subsectiontoc*` aligns the subsection's title to the other numbered subsections' titles; and `\subsectiontoc` not.

5.18.5/ Non-numbered Subsubsection in Table of Content

If you want to add a document subsubsection that has no subsubsection number but appearing inside the table of content, the classical LaTeX macros `\subsubsection` and `\subsubsection*` are inefficient. Indeed, `\subsubsection` is adding a numbered subsubsection inside the table of content, and `\subsubsection*` is adding an unnumbered subsubsection but not inside the table of content.

To add a unnumbered subsubsection inside the table of content, you could use one of the macros:
`\subsubsectiontoc[toctitle]{title}`
`\subsubsectiontoc*[toctitle]{title}`

The macros `\subsubsectiontoc` and `\subsubsectiontoc*` have the same effect except that `\subsubsectiontoc*` aligns the subsubsection's title to the other numbered subsubsections' titles; and `\subsubsectiontoc` not.

6

Package upmethodology-document

Version: 2013/09/29

The package upmethodology-document provides base function to manage document information (project, subproject, authors...).

6.1/ Document Information and Declaration

The informations associated to an UP document are:

- \theupmproject is the name of the project for which the document was produced;
- \theupmsubproject is the name of the sub-project for which the document was produced;
- \theupmdocname is the name of the document;
- \theupmdocref is the reference number of the document;
- \theupmfulldocname is the complete name of the document (composing by the project, subproject and name of the document).

You could declare the information about your document with one of the following functions:
\declaredocument{project}{name}{ref}
\declaredocumentex{project}{subproject}{name}{ref}
where the parameters are:

- project is the name of the project for which the document is for;
- subproject is the name of the sub-project for which the document is for;
- name is the name of the document;
- ref is the reference number of the document.

6.2/ Abstract and Key-words

You are able to declare the abstract and the key-words for your document. Both are basically used by the back page package.

6.2.1/ Declarations

The macro \setdocabstract is for entering the docment's abstract:
\setdocabstract[lang]{abstract_text}
where abstract_text is the text of your abstract and lang designates for which language the abstract text is for. If the language is not specified, this macro uses the current document language.

The macro \setdockeywords is for entering the docment's key-words:
\setdockeywords[lang]{keywords}
where keywords is the list of key-words and lang designates for which language the key-words are for. If the language is not specified, this macro uses the current document language.

6.2.2/ Rendering

The macro \theupmdocabstract is expanded with the abstract text:
\theupmdocabstract

The macro \theupmdockeywords is expanded with the key-words:
\theupmdockeywords

6.3/ Document Summary

You can obtain a document summary with the macro \upmdocumentsummary[width] which produces:

Document Summary	
Project	LaTeX Packages for Unified Process Methodology
Document	Official Documentation
Reference	UPM-2012-01
Version	28.4
Last Update	2013/10/09

6.4/ Change Icons

By default, this package uses the logo of *Arakhnê*.org as icons. You could change them with the macros:

- \defupmsmalllogo{filename} defines the small logo used in the headers for instance;
- \defupmlogo{filename} defines the logo used on the front page for instance.

The logos' filenames are accessible with the functions \theupmsmalldoclogo and \theupmdoclogo.

6.5/ Document Authors

An author is someone who participates to the writing of the document. You could register author identities with:
\addauthor[email]{firstname}{name}
\addauthor*[email]{firstname}{name}{comment}
\addauthorvalidator[email]{firstname}{name}
\addauthorvalidator*[email]{firstname}{name}{comment}

The list of the authors is accessible by two means:

- `\theauthorlist` is a coma-separated list of the authors' names;
- `\upmdocumentauthors` procudes an array of all the authors (see below for an example).

Authors		
Names	*Comments*	*Emails*
Stéphane Galland	Original Author	galland@arakhne.org

You could test if a string is the name of the author with:

- `\ifdocumentauthor{lowercasename}{then}{else}`; the first parameter **must** be lower case. If the `lowercasename` is the name of one of the authors, then the `then` clause is expanded, otherwise the `else` clause is expanded.

Authors		
Names	*Comments*	*Emails*
Stéphane Galland	Original Author	galland@arakhne.org

6.6/ Document Validators

A validator is someone who participates to the validation of the document. You could register validator identities with:
`\addvalidator[email]{firstname}{name}`
`\addvalidator*[email]{firstname}{name}{comment}`
`\addauthorvalidator[email]{firstname}{name}`
`\addauthorvalidator*[email]{firstname}{name}{comment}`

The list of the validators is accessible by two means:

- `\thevalidatorlist` is a coma-separated list of the validator' names;
- `\upmdocumentvalidators` procudes an array of all the validators (see below for an example).

Validators			
Names	*Comments*	*Emails*	*Initials*
Stéphane Galland	Original Author	galland@arakhne.org	

6.7/ Informed People

An informed people is someone who receives the document to be informed about its content. You could register informed people identities with:
`\addinformed[email]{firstname}{name}`
`\addinformed*[email]{firstname}{name}{comment}`

The list of the informed people is accessible by two means:

- `\theinformedlist` is a coma-separated list of the informed people' names;
- `\upmdocumentinformedpeople` procudes an array of all the informed people (see below for an example).

6.8/ Copyright and Publication Information

Package `upmethodology-document` provides several macros to define the copyright owner and the publication informations required to generate a publication page.

6.8.1/ Setting Information

Copyrighter is the people or the institution, or both, which is owning the copyright on the document. The following macro permits to set the identify of the copyighter in all the parts of the documents:
`\setcopyrighter{name}`

Publisher is the people or the institution, or both, which is publishing the document. Basically it is the same the copyrighter (see above):
`\setpublisher{name}`

Some times, copyright laws depend on the location where the document is printed. The following macro permits to put a message in the publication page which is indicating where the document is printed:
`\setprintingaddress{address}`

Publications may be identifier by international identifiers. Package `upmethodology-document` supports ISBN, ISSN and DOI: `\setisbn{number}`
`\setissn{number}`
`\setdoi{number}`

6.8.2/ Retreiving Information

The information set by the macros described in the previous section may be retreived with the following macros:
`\theupmcopyrighter`
`\theupmpublisher`
`\theupmprintedin`
`\theupmisbn`
`\theupmissn`
`\theupmdoi`

6.8.3/ Publication Page

The package `upmethodology-document` provides the `\upmpublicationpage` macro which is displaying a empty page with publication informations and optionally set the page number (default value is -1). Figure 6.1 illustrates the publication page of this document.

6.9/ Localization

The following macros defines some localized strings used by `upmethodology-document`:

- `\upm@lang@project`: Project;
- `\upm@lang@document`: Document;

This document describes the LaTeX Packages for Unified Process Methodology project.

TeX is a trademark of the American Mathematical Society.
`tex-upmethodology` is owned by Stéphane Galland, *Arakhnê*.org, France.

This document was realised with LaTeX and `tex-upmethodology`.

Reference : UPM-2012-01

Figure 6.1: Example of Publication Page generated with `\upmpublicationpage`

- \upm@lang@docref: Reference;
- \upm@lang@lastupdate: Last Update;
- \upm@lang@document@summary: Document Summary;
- \upm@lang@document@authors: Authors;
- \upm@lang@document@validators: Validators;
- \upm@lang@document@names: Names;
- \upm@lang@document@emails: Emails;
- \upm@lang@document@initials: Initials;
- \upm@lang@document@abstract: Abstract;
- \upm@lang@document@keywords: Key-words.

7

Package upmethodology-frontpage

Version: 2009/10/30

The upmethodology-frontpage package provides an front page for the UP documents. This package does not provides any public function. It is based on all the previous packages.

7.1/ Display the front page

The front cover is displayed by invoking one of the following macros:
\maketitle
\makefrontcover

7.2/ Change Front Page Layout

It is possible to change the layout of the front page with the macro:
\setfrontlayout{layout_name}
where layout_name must be one of:

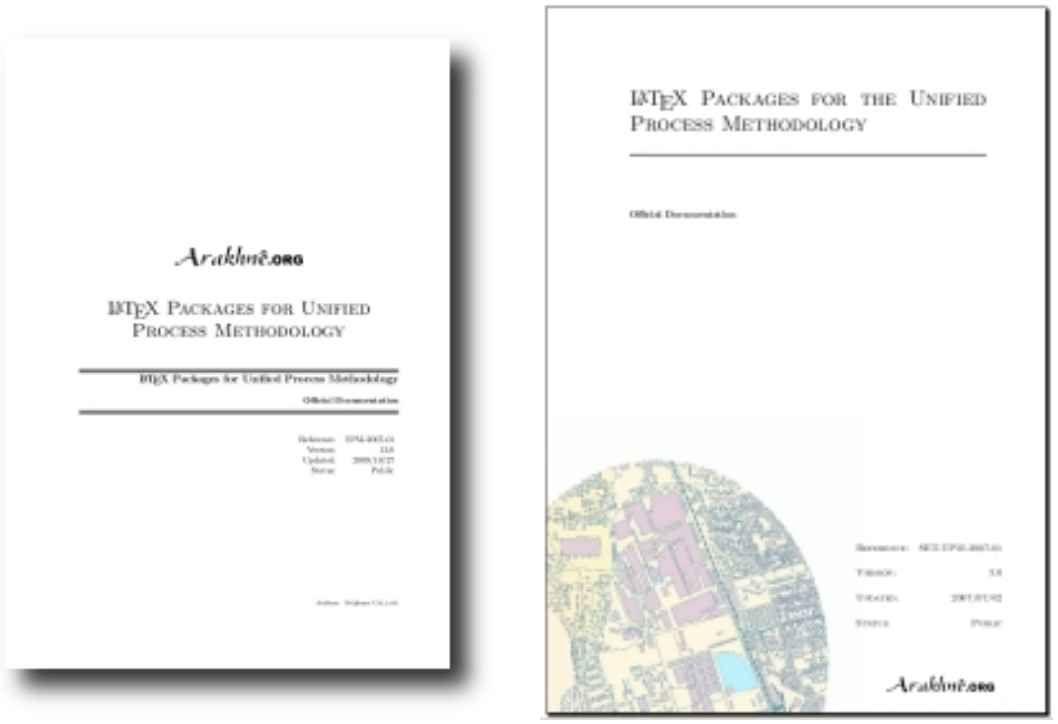
- classic: classic front page layout with title and logo;
- modern: front page layout with title and logo and background picture.

The figure 7.1 illustrates the differents layouts.

7.3/ Change Illustration Picture

It is possible to insert an illustration picture on the front page. You could specify the image with the macro:
\setfrontillustration[width_factor]{filename}
where:

- width_factor is the scaling factor of the picture according to the line width. If you specifies 1 the image will not be scaled, for .5 the image will be the half of its original width...
- filename is the name of picture to use as the illustration.



(a) `classic` (b) `modern`

Figure 7.1: Front Page Layouts

7.4/ Define a Front Page in Extensions

The `upmethodology-frontpage` package is able to use a page layout defined in a document extension (see chapter 9 for details on document extension).

A LaTeX macro must be defined in the `upmext-NAME.cfg` file of the extension. The name of this macro (for example `mylayout`) must be set with the `\set` macro in the same file:
`\set{frontpage}{mylayout}`

7.5/ Localization

The following macros defines some localized strings used by `upmethodology-frontpage`:

- `\upm@lang@front@authors`: Authors;

8

Package upmethodology-backpage

Version: 2009/10/30

The package upmethodology-backpage provides an back page for the UP documents. This package does not provides any public function. It is based on all the previous packages.

8.1/ Display the back page

The back cover is displayed by invoking the following macro:
\makebackcover

8.2/ Change Back Page Layout

It is possible to change the layout of the back page with the macro:
\setbacklayout{layout_name}
where layout_name must be one of:

- none: no back page.

8.3/ Define a Back Page in Extensions

The upmethodology-backpage package is able to use a page layout defined in a document extension (see chapter 9 for details on document extension).

A LaTeX macro must be defined in the upmext-NAME.cfg file of the extension. The name of this macro (for example mylayout) must be set with the \set macro in the same file:
\set{backpage}{mylayout}

9

Package upmethodology-extension

Version: 2012/09/21

The package upmethodology-extension provides tools to create layout and rendering extensions. It is possible to write an extension to the upmethodology-document package. An extension is able to override several values from the default upmethodology-packages or may be used by the other suite's packages. For example, the Systems and Transport laboratory[I(11)] extension is providing laboratory's icons, publisher's name and page layouts.

9.1/ Load a Document Extension

To load and use a document extension, you must invoke the macro:
\UseExtension{extension_name}
where extension_name is the identifier of the extension to load. The extension's files must be inside your LaTeX search path.

9.2/ Write a Document Extension

A document extension could be written and described inside a file named upmext-NAME.cfg, where NAME is the name of the extension. This file must be put in your LaTeX search path.

The upmext-NAME.cfg file is a LaTeX file in which a set of definition macros are put. These macros must respect the LaTeX syntax.

The \DeclaraCopyright macro permits to declare several copyright information about the extension:
\DeclareCopyright[lang]{extension_name}{year}{copyrighter}{trademark and copyright information}

This macro declares the copyright value which contains the copyright text (for this documentation ""). This macro also declares the trademarks value which contains the trademark and other related informations about the extension (for this documentation "").

Additional macros are provided to redefine the upmethodology-document constants:
\Set[lang]{variable_name}{value}

The variable_name is the name of the value to override. It must be taken in one of the names listed in table 9.1. The lang parameter is a language identifier. It is used to restrict the definition to a specific language. If not given, the default language is used insteed. The image_name and image_scale are the name of the image file and the scaling factor respectively.

Value Name	*Description*
logo	the filename of the picutre which must be used as a large logo.
smalllogo	the filename of the picutre which must be used as a small logo.
copyrighter	the name of the authors or the institution which own the copyright on the document.
publisher	the name of the document's publisher. The `lang` parameter is supported.
printedin	the location/address where this document is printed.
frontillustration	the image to use as illustration. The `lang` parameter is ignored.
frontpage	the name of the front page style — not the LaTeX macros — to layout the front page. OR the front page illustration.
backpage	the LaTeX macros to layout the back page. OR the back page illustration.
cfrontpage	the LaTeX macros — not the name of the front page style — to layout the front page.

Table 9.1: List of overiddable value names

The `\Get` macro permits to retreive the value defined by a `\Set`:
`\Get{variable_name}`

The `\Unset` macro permits to remove the definition of a value:
`\Unset{variable_name}`

The `\Ifnotempty` macro permits to expand the LaTeX macros if the given text is not empty:
`\Ifnotempty{text}{latex_code}`

The `\Ifempty` macro permits to expand the LaTeX macros if the given text is empty:
`\Ifempty{text}{latex_code}`

The `\Ifelsedefined` macro permits to expand the LaTeX macros in `then_code` if a value with the given name was defined, or to expand the LaTeX macros in `else_code` if no value with the given name was defined:
`\Ifelsedefined{value_name}{then_code}{else_code}`

The `\Put` macro is an extension of the standard picture `\put` macro. It takes into account the joint margin applied in two sided documents when it is used on page's backside (eg. the back page of the document):
`\Put(x,y){macros}`

This macro must be used inside a `picture` environment in place of the standard `\put` macro.

10

Package upmethodology-task

Version: 2009/10/30

The LaTeX package `upmethodology-task` provides a set of macros to define project's tasks.

During LaTeX compilation this package could log the message `"Project Task(s) may have changed. Rerun to get cross-references right"` when some task information was not found or due to cross-references on them.

10.1/ Task Definition

The definition of a task could be made only inside one of the following environments:
`\begin{taskdescription}{id}...\end{taskdescription}`
`\begin{taskdescription*}{id}...\end{taskdescription*}`
where `id` is the identifier of the task.

The environment `taskdefinion` displays the task's description with a call to `\thetaskdescription{id}`. In the opposite `taskdefinition*` never displays the ta's description.

Inside one of the task's definition environment above, you could use one of the following macros to define the task's attributes:

- `\taskname{name}`
  permits to defines the name of the task;
- `\tasksuper{id}`
  indicates that the current task is a sub-task of the task identified by the given identifier;
- `\taskcomment{text}`
  permits to describe the task's purposes and goals (will be shown in the description box of the task's description);
- `\taskprogress{percent}`
  defines the percent for thtask's archieving;
- `\taskstart{date}`
  permits to set the starting date of the task (real or predicted);
- `\taskend{date}`
  permits to set the finished date of the task (real or predicted);
- `\taskmanager{name}`
  adds a task's manager into the list of the managers;
- `\taskmember{name}`
  adds a task's member into the list of the members;

- \taskmilestone{date}{comment}
  add a milestone into the task for the given date and described by the given comment.

10.2/ Task Reference

You could reference any information about the defined tasks in your document. In case you used cross-references this package could log the message "Project Task(s) may have changed. Rerun to get cross-references right" to complain about rebuilding of our document.

The following macros are available:

- \thetasksuper{id}
  replies the identifier of the parent task corresponding to the task identified by id;
- \thetaskname{id}
  replies the name of the the task identified by id;
- \thetaskcomment{id}
  replies the description for the the task identified by id;
- \thetaskprogress{id}
  replies the archieving percent for the the task identified by id;
- \thetaskstart{id}
  replies the starting date for the the task identified by id;
- \thetaskend{id}
  replies the ending date for the the task identified by id;
- \thetaskmanagers{id}
  replies the managers' list for the the task identified by id;
- \thetaskmembers{id}
  replies the members' list for the the task identified by id;
- \thetaskmilestones{id}
  replies the list of milestone's dates for the the task identified by id;
- \thetaskmilestonecomment{id}{date}
  replies the comment of the given milestone for the the task identified by id;
- \thetaskdescription[width]{id}
  replies the complete description of the the task identified by id.

10.3/ Localization

The following macros defines some localized strings used by upmethodology-task:

- \upm@task@lang@task: Task;
- \upm@task@lang@escription: Description;
- \upm@task@lang@startat: Start at;
- \upm@task@lang@endat: End at;
- \upm@task@lang@archieved: Archieved;
- \upm@task@lang@managers: Managers;

- \upm@task@lang@members: Members;
- \upm@task@lang@Milestones: Milestones;
- \upm@task@lang@subtask: Sub-task of.

11

PACKAGE UPMETHODOLOGY-CODE

Version: 2009/10/30

The LaTeX package `upmethodology-code` provides a set of macros for source code formatting. The supported source codes are UML, Java and C++.

You could load the package with the following options:

`uml`	use the UML notation (default value)
`java`	use the Java notation
`cpp`	use the C++ notation

You could also change the notation language with the macro:
`\upmcodelang{upm|java|cpp}`

The provided macros are listed in the following table:

macro	UML	Java	C++
Prototypes			
`\jclass{TheClass}`	TheClass	TheClass	TheClass
`\jinterface{TheInterface}`	*TheInterface*	*TheInterface*	*TheInterface*
`\jpackage{ThePackage}`	ThePackage	ThePackage	ThePackage
`\jfunc{FunctionName}`	`FunctionName`	`FunctionName`	`FunctionName`
Types			
`\jclazz`	**class**	**Class**	**class**
`\jvoid`	**void**	**void**	**void**
`\jboolean`	**boolean**	**boolean**	**bool**
`\jint`	**integer**	**int**	**int**
`\jlong`	**long integer**	**long**	**long**
`\jfloat`	**float**	**float**	**float**
`\jdouble`	**double**	**double**	**double**
`\jchar`	**character**	**char**	**char**
`\jstring`	**string**	String	std::string
`\jarray{T}`	**array of** Ts	T[]	T[]
`\jcollection{T}`	**collection of** Ts	Collection <T>	std::vector <T>
`\jset{T}`	**set of** Ts	Set <T>	std::set <T>

macro	UML	Java	C++
Constants			
\jtrue \jfalse	TRUE FALSE	TRUE FALSE	TRUE FALSE
Operations			
\jcode{source code} \jcall{fct}{params} \jop{operator}	source code fct(params) operator	source code fct(params) operator	source code fct(params) operator

12

Authors and License