Preparing Proposals in LaTeX with `proposal.cls`

Michael Kohlhase
Computer Science, Jacobs University Bremen
http://kwarc.info/kohlhase

January 7, 2015

Abstract

The `proposal` class supports many of the generic elements of Grant Proposals. It is optimized towards collaborative projects, and should specialized to particular funding agencies.

Contents

1   Introduction

Writing grant proposals is a collaborative effort that requires the integration of contributions from many individuals. The use of an ASCII-based format like LaTeX allows to coordinate the process via a source code control system like SUBVERSION, allowing the proposal writing team to concentrate on the contents rather than the mechanics of wrangling with text fragments and revisions.

The `proposal` class supports many of the generic elements of Grant Proposals. The package documentation is still preliminary, fragmented and incomplete.

The `proposal` class is distributed under the terms of the LaTeX Project Public License from CTAN archives in directory `macros/latex/base/lppl.txt`. Either version 1.0 or, at your option, any later version. The CTAN archive always contains the latest stable version, the development version can be found on GitHub at https://github.com/KWARC/LaTeX-proposal. For bug reports please use the issue tracker there.

2   The User Interface

In this section we will describe the functionality offered by the `proposal` class along the lines of the macros and environments the class provides.

2.1   Package Options

The `proposal` package takes the options `submit`, `noworkareas`, `RAM`, `deliverables`, `wpsubsection`, `keys`, `svninfo`, `gitinfo`, and `public`.

submit   The `submit` option will disable various proposal management decorations which are enabled by default for submission.

noworkareas   The `noworkareas` option specifies that we do not want to structure our work plan into work areas (see section 2.5).

RAM   The `RAM` option specifies that we specify research assistant months in the effort tallies (see section 2.5).

deliverables   The `deliverables` option specifies that we specify deliverables in the grant proposal (see section 2.9). As the deliverables management needs extra support, we only activate them via this option.

wpsubsection   The `wpsubsection` option specifies that we want to see subsections headings for the WPs (and WAs, if we have them).

report   The `report` option specifies that we want to use the `report.cls` class as a basis for `proposal` instead of the default `article.cls`.

keys   The `keys` option specifies that we want to see the values of various keyval arguments in the margin.

svninfo   The `svninfo` option specifies specifies that we want to use the `svninfo` package for displaying version control metadata in the document (except when the `submit` option is also given). For this we need the `svninfo` metadata line of the form

```
\SVN $Id: proposal.tex 13610 2007-07-11 04:30:16Z kohlhase $
\svnKeyword $HeadURL: https://svn.kwarc.info/../proposal.tex $
```

at the beginning of each file (or in the preamble).

gitinfo   Analogously, the `gitinfo` option uses the `gitinfo2` package for GIT metadata. Note that you will need to install the post-commit hooks in your working copy according to [Lon] for this to work.

public   Finally, the `public` option allows to hide certain sensitive (e.g. financial) parts of the proposal. private   For this, the `proposal` class provides the `private` environment. If the option `public` is set, the parts of the document between `\begin{private}` and `\end{private}` do not produce output.

This is useful for producing public versions of the proposal that hide confidential parts. Note that both `\begin{private}` and `\end{private}` *have to be on lines of their own may not have any leading whitespace* otherwise an error occurs and LATEX gives error messages that are difficult to comprehend. An alternative way to distinguish private and public sections are to use the `\ifpublic` conditional: `\ifpublic{3}\else{5}\fi` will result in "5" in the submitted draft and "3" in the public document.

2.2 Proposal Metadata and Title page

The metadata of the proposal is specified in the `proposal` environment, which also generates the title page and the first section of the proposal as well as the last pages of the proposal with the signatures, enclosures, and references. The `proposal` environment should contain all the mandatory parts of the proposal text. The `proposal` environment uses the following keys to specify metadata.

- `title` for the proposal title (used on the title page),
- `instrument` for the instrument of funding that you would like to apply for,
- `acronym` for the proposal acronym, possibly accompanied by an `acrolong` that explains it. The acronym will also be used in the page headings.
- `start` for the start date of the proposed fragment of the project, and `months` for the length of the proposal in months. Both have to be specified for the `proposal` class to work.
- If the proposal only concerns a part of a longer-running project, the `since` key allows to specify the date since when the overall project runs. Finally, the `fundsuntil` allows to specify a date until which the funds last.
- `discipline` for the academic discipline and `areas` for the research areas in that discipline.
- `PI` to declare the principal investigator. For collaborative proposals we can use the `PI` key multiple times. The `proposal` package uses the `workaddress` package for representation of personal metadata, see [Koh14c] or the file `proposal.tex` for details.
- Many collaborative proposals are shared between two institutions, which we can declare with the `site` key. As this changes the interface this should not be used for single-institution proposals. We will describe the setup for a single-site proposal below and point out the differences. The example `proposal.tex` is a two-site proposal.

If the `acronym` and `acrolong` are given, then they automatically define the macros `\pn` and `\pnlong` which allow to use the project acronym (<u>p</u>roject <u>n</u>name) and its long version in the text. Note that these macros use `\xspace` internallly, so they do not have to be enclosed in curly braces.

2.3 Proposal Appearance

The `proposal` environment takes a second set of keyval arguments that allow to fine-tune the appearance of the proposal document. [1]

- If the `compactht` key is given (it does not need a value), then the header tables[2] are made compact, i.e. the sites that do not have a contribution to the work package or work area do not get listed. This is useful for proposals with more than 8 partners.

The `proposal` package supplies the `emphbox` environment to create boxes of emphasized material we want to call attention to.

2.4 Objectives

The work plan starts with a discussion of objectives, which may be referenced in the text later. The `proposal` package provides the `objective` environment that allows to mark up individual objectives. It takes a keyval argument with the keys `id` for identification, `title` for the objective title, and `short` for a short title that can be used for referencing when the title is too long. The objectives can be referenced via `\OJBref{`⟨*id*⟩`}` by their label and via `\OJBtref{`⟨*id*⟩`}` by label

---

[1] EDNOTE: move the RAM, wpsectionheadings,... options here.
[2] EDNOTE: describe them somewhere and reference here

and (short if it was specified) title.

2.5 Work Areas and Work Packages

Grant proposals have another part that is often highly stylized; the work plan. This is usually structured into "work packages" — i.e. work items that address a cohesive aspect of the proposed work. These work packages are usually consecutively numbered, have a title, and an associated effort estimation. As work packages are the "atomic" planning units, they are usually heavily cross-referenced. A well-written proposal usually contains a table giving an overview over the work packages and their efforts and a Gantt chart showing the temporal distribution of the proposed work to allow the reviewers to get a clear picture of the feasibility of the research and development proposed. But this picture is also essential during the development of a proposal (which the `proposal` package aims to support), when the work packages (and their estimated efforts) usually change considerably. Therefore the `proposal` class standardizes markup for work packages and automatically computes the work package table (which can be inserted into the table via the `\wpfig` macro) and the Gantt Chart (see Section 2.8).

`\wpfig`

`workplan`   To achieve the automation, work plan is marked up by the `workplan` environment, which sets up various internal counters and bookeeping macros. It contains texts and `workpackage` environments for the work packages.

`workpackage`   The purpose of the `workpackage` environment is to mark up a fragment of text as a work package description and specify the metadata so that it can be used in the work package table and Gantt chart generation. The metadata is specified by the following keys:

`id`   • The `id` key is used to specify a label for cross-referencing the work package or work group, it must be document-unique.

`title`   • The `title` and `short` keys are used for the work package/group title. The short title is used
`short`   in tables and should not be longer than 15 characters.

`wphases`   • The `wphases` key is used according to Section 2.7

`requires`   • The `requires` key can be used to mark, up dependencies between tasks. If `requires=\taskin{⟨rid⟩}{⟨wp⟩}` is given in a task with `id=⟨t⟩`, then task ⟨rid⟩ in work package ⟨wp⟩ must be completed for task ⟨t⟩ to become possible. This key will draw an arrow into the gantt chart from the end of task ⟨rid⟩ to ⟨t⟩. Note that dependencies should always point forward in time. Furthermore, note that the fact that dependencies always go from the end of the source to the beginning of the target work phase is intentional, if this does not meet your needs, then you should probably break a work phase into pieces that can be addressed separately.

`RM`   • In single-site proposals, the `RM` (and `RAM` if the `RAM` option was given) keys are used to specify
`RAM`   the estimated efforts to be expended on research and development in this work package. Both are specified in person months. `RM` is used for "researcher months" (wissenschaftlicher Mitarbeiter) and `RAM` for "research assistant months" (wissenschaftliche Hilfskraft).

`*RM`   • In multi-site proposals, the `proposal` package generates the keys ⟨site⟩RM (and ⟨site⟩RAM)
`*RAM`   where ⟨site⟩ is any site label declared via the `site` key in the top-level `proposal` environment. This can be used to specify the person months that the site spends on this work package (the value for work groups is automatically computed (remember to run LaTeX twice for this)).

`lead`   • In multi-site proposals the `lead` key specifies the work package or work group lead, the value of this feature should be the short name of the respective partner.

It is often useful to group the work packages in a proposal further (especially for larger, collab-
`workarea` orative proposals). This can be done via the `workarea` environment, which groups work packages. This environment takes the same keys as the `workpackage` environment, except for the efforts, which can be computed automatically from the work packages it groups.

As the author of the `proposal` class likes more structured proposals, using work areas is the default, but the `proposal` class can also be used with the `noworkareas` option for less structured (smaller) proposals.

4

2.6 Tasks

tasklist In the work packages we can list tasks that need to be undertaken with the `tasklist` environment.
task The individual tasks are marked up with the `task` environment. This takes a keyval argument with the keys `id` for identification, `title` for a title, and the workphase keys `wphrases`, `start`, `end`, and `force` (see Section 2.7). For planning involvement we can specify the overall person months via the PM key, the task lead via `lead`, and the partners involved via the `partners` key. Finally task dependencies can be specified via the `requires` key.

\taskref Tasks can be referenced by the `\taskref` macro that takes two arguments: the work package identifier and the task identifier. As for work packages and work areas, there is a long reference
\tasktref variant with work package title: `\tasktref`. Finally, `\localtaskref` references a task in the local
\localtaskref work package by the identifier in its argument.

2.7 Work Phase Metadata

wphases The `task` and `workpackage` allow the `wphases` key to specify the a list of work phases. The value of this key is comma-separated list of work phase specifications of the form $\langle start\rangle$-$\langle end\rangle$ or $\langle start\rangle$-$\langle end\rangle$!$\langle force\rangle$, where $\langle start\rangle$ and $\langle end\rangle$ delimit the run time of the work phase and the optional !$\langle force\rangle$ specifies the work force, i.e. the intensity of work as a number between 0 and 1. If no force is given, the default is 1. The main reason for specifying this metadata for tasks is to generate a Gantt chart (see Section 2.8).

2.8 Gantt Charts

Gantt charts are used in proposals to show the distribution of activities in work packages over time.
gantt A gantt chart is represented by the `gantt` environment that takes a on optional keyval argument.
xscale The keys `xscale` and `yscale` are used to specify a scale factors for the chart so that it fits on the
yscale page. The `step` key allows to specify the steps (in months) of the vertical auxiliary lines. Finally,
step the `draft` key specifies that plausibility checks (that can be expensive to run) are carried out.
draft Note that the value does not have to be given, so `\begin{gantt}{draft,yscale=.5,step=3}` is a perfectly good invocation.

\ganttchart Usually, the `gantt` environment is not used however, since it is part of the macro that takes the same keys. This generates a whole Gantt chart automatically from the work phase specifications in the work packages. As above we have to run LaTeX two times for the work phases to show up.

2.9 Milestones and Deliverables

Many proposal formats foresee that project progress will be tracked in the form of *milestones* – points in the project, where a predefined state of affairs is reached – and *deliverables* – tangible project outcomes that have to be delivered. Correspondingly, milestones and deliverables have to be specified in the proposal and accounted for in the project reports. To facilitate this the `proposal` class and its instances provide a simple infrastructure for dealing with milestones and deliverables.

milestones Milestones are usually given in a special table[1], which we markup up with the `milestones` environment that takes care of initialization and numbering issues. This contains a list of milestone
\milestone descriptions via the `\milestone` macro which is invoked as `\milestone[`$\langle keys\rangle$`]{`$\langle title\rangle$`}{`$\langle desc\rangle$`}`, where $\langle keys\rangle$ supports the keys `id` for identification `month` for specifying the milestone date (in months of the project duration), and `verif` for specifying a means of verification[2] Mile-
\milestone@label stones are numbered with labels whose shape can be customized by redefining `\milestone@label`
\mileref and referenced by the `\mileref{`$\langle id\rangle$`}` and `\miletref{`$\langle id\rangle$`}` for a reference with milestone title.
\miletref `\pdatacount{all}{miles}` gives the number of milestones.

---

[1]this is the default provided by the base `proposal` class, it can be specialized for proposal class instances by redefining the `@milestones` environment and correspondingly the `milestone` macro.

[2]Arguably, this set of keys is inspired by EU proposals, but can be extended in class instances.

Deliverables are usually defined as part of the work package descriptions (see Section 2.5) and listed in an overview table in a separate of the proposal. As for the milestones, we use an environment `wpdelivs` that contains the deliverable descriptions. These are marked up via the environment which takes an optional keyval argument for the deliverable metadata a regular argument for the title and contains the description of the deliverable as the body. For the metadata we have the keys `id` for the deliverable identifier, `due` for the target date (a number that denotes the project month), `nature` and `dissem` for specifying the deliverable nature and dissemination status (usually as short strings prescribed by the proposal template), and `miles` for the milestone this deliverable is targeted for (specified by the milestone identifier). For repeating deliverables (e.g. project reports), both `due` and `miles` can contain comma-separated lists. Deliverables are numbered by labels whose shape can be customized by number, where the shape of the label can be specified by redefining `\deliv@label` and referenced by `\delivref{`$\langle wp\rangle$`}{`$\langle id\rangle$`}` where $\langle wp\rangle$ is the work package identifier and $\langle id\rangle$ that if the deliverable and `\delivtref{`$\langle wp\rangle$`}{`$\langle id\rangle$`}` for a reference with title. `\pdatacount{`$\langle wp\rangle$`}{delivs}` gives the number of milestones of the work package $\langle wp\rangle$ `\pdatacount{all}{delivs}` that of all deliverables (aggregating over all work packages).

Some proposal templates ask for an overview table of the deliverables which aggregates the deliverables of the respective work packages and areas ordered by due date. This can be generated with the `\inputdelivs` macro. This works index generation in LaTeX. The `wpdeliv` environment writes the deliverable data to a file $\langle main\rangle$`.delivs`, which can be processed externally (usually just sorting with `sort` in Unix is sufficient) into $\langle main\rangle$`.deliverables`, which is then input via the `\inputdelivs` macro.

In some proposals, also work areas can have deliverables, then the above hold analogously for `wpdelivs` and `wadeliv` environments.

Note that handling deliverables adds considerable overhead to proposal formatting and adds auxiliary files, so they are only activated if the `deliverables` option is given (see Section 2.1).

Margin notes: wpdelivs, wpdeliv, \deliv@label, \delivref, \delivtref, \inputdelivs, wadelivs, wadeliv

2.10 Referencing and Hyperlinking

The `proposal` package extends the hyperlinking provided by the `hyperref` package it includes to work packages, work groups, ... Whenever these are defined using the `proposal` infrastructure, the class saves the relevant information in the auxiliary file $\langle proposal\rangle$`.aux`. This information can be referenced via the `\pdataref` macro, which takes three arguments.

In a reference `\pdataref{`$\langle type\rangle$`}{`$\langle id\rangle$`}{`$\langle aspect\rangle$`}` the first argument $\langle type\rangle$ specifies the type of the object (currently one of `wp`, `wa`, and `partner`) to be referenced, $\langle id\rangle$ specifies the identifier of the referenced object (it matches the identifier given in the `id` key of the object), and $\langle aspect\rangle$ specifies the aspect of the saved information that is referenced.

For a partner $\langle aspect\rangle$ can be one of `number` (partner number), `short` (partner acronym), `long` (official partner name), `nationality` (partner nationality).

For a work package $\langle aspect\rangle$ can be `number`, (the work package number), `label` (the label **WP**n where n is the work package number for referencing), `title` (the work package title), `lead` the work package leader, `short` (a short version of the WP title for tables). For work groups we have the same aspects with analogous meanings. In all cases, the referenced information carries a hyperlink to the referenced object.

The `\pdataRef` macro is a variant of `\pdataref` that also carries a hyperlink (if the `hyperref` package is loaded).

The `\pdatacount` macro gives access to the numbers of certain aspects. For instance, the number of work packages in the proposal can be cited by `\pdatacount{all}{wp}`, similarly for work areas (if they are enabled), and finally, `\pdatacount{`$\langle wa\rangle$`}{wp}` gives the number of work packages for a work area $\langle wa\rangle$. This is very useful for talking about work plans in a general way. Other objects that can be counted are deliverables (`\pdatacount{all}{deliverables}`) and milestones (`\pdatacount{all}{milestones}`).

Note that since the referencable information is written into the project data file $\langle proposal\rangle$`.pdata` file, it is available for forward references. However, it will only become available when the project

Margin notes: \pdataref, \pdataRef, \pdatacount

data file is read, so the proposal has to be formatted twice for references to be correct.

Finally, the `proposal` package supplies specialized reference macros for work packages and areas. The `\WPref` macro takes a work package identifier as an argument and makes a reference: `\WPref{⟨id⟩}` abbreviates `\pdataRef{wp}{⟨id⟩}{label}`. The `\WPtref` macro is similar, but also prints out the (short) title: `\WPref{⟨id⟩}` abbreviates `\pdataRef{wp}{⟨id⟩}{label}`: `\pdataRef{wp}{⟨id⟩}{title}`. Unless the `noworkareas` macro is set, we also have the variants `\WAref` and `\WAtref` for work areas.

(margin: `\WPref` `\WPtref` `\WAref` `\WAtref`)

2.11 Coherence

Many proposals require ways to show coherence between the partners. The `proposal` class offers the macro `\coherencematrix` for this which generates a matrix of symbols specifying joint publications and joint projects by the project partners that have been declared by the `\jointpub`, `\jointproj`, and `\jointorga` macros before. These macros all take a comma-separated list of site identifiers as an argument. Use for instance `\jointproj{a,b,c}` to specify that the sites with the identifiers `a`, `b` and `c` have a joint project. `\coherencetable` is a variant which packages the coherence table in a table figure with label `tab:collaboration`.

The symbols used an be configured by redefining `\jpub`, `\jproj`, and `\jorga`.

(margin: `\coherencematrix` `\jointpub` `\jointproj` `\jointorga` `\coherencetable` `\jpub` `\jproj` `\jorga`)

2.12 Localization

The `proposal` class offers some basic support for localization. This is still partial though, and I am not sure that this is the best way of setting things up. What I do is to define macros for all generated texts that can be redefined in the proposal classes that build in `proposal`. For instance the `dfgproposal` class [Koh14b] provides an option `german` for german-language proposals and project reports that triggers a redefinition of all of these macros at read time.

3 Limitations and Enhancements

The `proposal` is relatively early in its development, and many enhancements are conceivable. We will list them here.

1. macros cannot be used in work package and work area titles. They really mess up our `\wpfig` automation. The problem is that they are evaluated too early, and our trick with making them undefined while collecting the parts of the table-rows only works if we know which macros we may expect. We might specify all "allowable" macros in an optional key `protectmacro`, which is defined via

   ```
   \define@key{wpfig}{protectmacro}{\epandafter\let\csname #1\endcsname=\relax}
   ```

   But I am not sure that this will work.

2. It would be great, if in the Gantt Charts, we could include some plausibility checks (for draft = not `submit` mode). I can see two at the moment:

   - calculating the effort (i.e. the weight of the black area) and visualizing it. Then we could check whether that is larger than the effort declared for the work package.
   - calculating (and visualizing) the monthly effort. That should be kind of even (or it has to be explained in the positions requested).

3. we currently do not have a way to relate `PI`s to `site`s, but we do not really need to.

If you have other enhancements to propose or feel you can alleviate some limitation, please feel free to contact the author.

Acknowledgements

4 The Implementation

In this section we describe the implementation of the functionality of the `proposal` package.

4.1 Package Options and Format Initialization

We first set up the options for the package.

```
1 ⟨*cls | reporting⟩
2 \newif\if@wpsubsection\@wpsubsectionfalse
3 \newif\ifsubmit\submitfalse
4 \newif\ifpublic\publicfalse
5 \newif\ifkeys\keysfalse
6 \newif\ifdelivs\delivsfalse
7 \newif\ifwork@areas\work@areastrue
8 \newif\if@RAM\@RAMfalse
9 \newif\if@svninfo\@svninfofalse
10 \newif\if@gitinfo\@gitinfofalse
11 \def\proposal@class{article}
12 \DeclareOption{wpsubsection}{\@wpsubsectiontrue}
13 \DeclareOption{submit}{\submittrue}
14 \DeclareOption{gitinfo}{\@gitinfotrue}
15 \DeclareOption{svninfo}{\@svninfotrue}
16 \DeclareOption{public}{\publictrue}
17 \DeclareOption{noworkareas}{\work@areasfalse\PassOptionsToClass{\CurrentOption}{pdata}}
18 \DeclareOption{RAM}{\@RAMtrue}
19 \DeclareOption{report}{\def\proposal@class{report}}
20 \DeclareOption{keys}{\keystrue}
21 \DeclareOption{deliverables}{\delivstrue}
22 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
23 \ProcessOptions
```

Then we load the packages we make use of

```
24 \LoadClass[a4paper,twoside]{\proposal@class}
25 \RequirePackage{amssymb}
26 \RequirePackage{url}
27 \RequirePackage{graphicx}
28 \RequirePackage{colortbl}
29 \RequirePackage{xcolor}
30 \RequirePackage{rotating}
31 \RequirePackage{fancyhdr}
32 \RequirePackage{array}
33 \RequirePackage{xspace}
34 \RequirePackage{comment}
35 \AtBeginDocument{\ifpublic\excludecomment{private}\fi}
36 \RequirePackage{tikz}
37 \RequirePackage{paralist}
38 \RequirePackage{a4wide}
39 \RequirePackage{boxedminipage}
40 % so that ednotes in wps do not run out of symbols
41 \renewcommand{\thempfootnote}{\roman{mpfootnote}}
42 \renewcommand{\familydefault}{\sfdefault}
43 \RequirePackage[scaled=.90]{helvet}
44 \RequirePackage{textcomp}
45 \RequirePackage[hyperref=auto,style=numeric,defernumbers=true,backend=bibtex,backref=true,firstinits=true,max
46 \RequirePackage{csquotes}
47 \RequirePackage{mdframed}
48 \RequirePackage{pdata}
```

in submit mode, we make the links a bit darker, so they print better.

```
49 \definecolor{darkblue}{rgb}{0,0,.7}
50 \ifsubmit\def\prop@link@color{darkblue}\else\def\prop@link@color{blue}\fi
51 \RequirePackage[bookmarks=true,linkcolor=\prop@link@color,
52   citecolor=\prop@link@color,urlcolor=\prop@link@color,colorlinks=true,
53   breaklinks=true, bookmarksopen=true]{hyperref}
```

the `ed` package [Koh14a] is very useful for collaborative writing and passing messages between collaborators or simply reminding yourself of editing tasks, so we preload it in the class. However, we only want to show the information in draft mode. Furthermore, we adapt the options for the `svninfo` and `gitinfo2` packages.

```
54 \ifsubmit
55 \RequirePackage[hide]{ed}
56 \if@svninfo\RequirePackage[final,today]{svninfo}\fi
57 \else
58 \RequirePackage[show]{ed}
59 \if@svninfo\RequirePackage[eso-foot,today]{svninfo}\fi
60 \if@gitinfo\RequirePackage[mark]{gitinfo2}\fi
61 \fi
62 \renewcommand\ednoteshape{\sl\footnotesize}
```

private   We configure the `comment` package, so that it provides the `private` environment depending on the status of the `public` option.

```
63 \ifpublic\excludecomment{private}\else\includecomment{private}\fi
```

And we set up the appearance of the proposal. We want numbered subsubsections.

```
64 \setcounter{secnumdepth}{3}
```

We specify the page headings.

```
65 \newif\ifofpage\ofpagefalse
66 \fancyhead[RE,LO]{\prop@gen@acronym}
67 \fancyhfoffset{0pt}
68 \fancyfoot[C]{}
69 \newcommand\prop@of@pages[2]{page~#1\ifofpage~of~#2\fi}
70 \fancyhead[LE,RO]{\prop@of@pages\thepage{\pdataref@num{prop}{page}{last}}}
71 \pagestyle{fancyplain}
72 ⟨/cls | reporting⟩
```

4.2   Proposal Metadata

pdata   Most of the metadata functionality is encapsulated into the `pdata` package, which is shared by the proposal and report classes. `pdata.sty` first loads the `workaddress` package from sTeX and supplies the Euro symbol.

```
73 ⟨∗pdata⟩
74 \RequirePackage{workaddress}[2011/05/03]
75 \RequirePackage{eurosym}
```

We define the keys for metadata declarations in the `proposal` environment, they park their argument in an internal macro for use in the title page. The `site` key is the most complicated, so we take care of it first: We need a switch `\if@sites` that is set to true when the `site` key is used. Furthermore `site=⟨site⟩` makes new keys ⟨site⟩RM and ⟨site⟩RAM (if the `RAM` option was set) for the `workpackage` environment and records the sites in the `\prop@gen@sites` token register.

```
76 \newif\if@sites\@sitesfalse\let\prop@gen@sites=\relax%
77 \newcounter{@site}%
78 \define@key{prop@gen}{site}{\@sitestrue\@dmp{site=#1}%
79 \stepcounter{@site}\pdata@def{site}{#1}{number}{\the@site}%
80 \@ifundefined{prop@gen@sites}{\xdef\prop@gen@sites{#1}}{\xdef\prop@gen@sites{\prop@gen@sites,#1}}%
81 \define@key{prop@gen}{#1RM}{\pdata@def{site}{#1}{intendedRM}{##1}}%
```

```
82 \if@RAM\define@key{prop@gen}{#1RAM}{\pdata@def{site}{#1}{intendedRAM}{##1}}\fi
83 \define@key{workpackage}{#1RM}{\pdata@def\wp@id{#1}{RM}{##1}}%
84 \if@RAM\define@key{workpackage}{#1RAM}{\pdata@def\wp@id{#1}{RAM}{##1}}\fi
85 \define@key{prop@gen}{#1employed}{{\let\tabularnewline\relax\let\hline\relax\let\wa@ref\relax%
86 \@ifundefined{prop@gen@employed@lines}%
87 {\xdef\prop@gen@employed@lines{\wa@ref{institution}{#1}{shortname} & ##1\tabularnewline\hline}}%
88 {\xdef\prop@gen@employed@lines{\prop@gen@employed@lines \wa@ref{institution}{#1}{shortname} & ##1\tabularnewl
```

If there are no sites, then we have to define keys `RM` and `RAM` that store the intended research (assistant months). Unfortunately, we cannot just include this in the `\if@sites` conditional here, since that is only set at runtime.

```
89 \define@key{prop@gen}{RM}{\@dmp{RM=#1}\if@sites%
90 \PackageWarning{Do not use the RM key in the presence of sites}\else%
91 \pdata@def{all}{intended}{RM}{#1}\fi}
92 \define@key{prop@gen}{RAM}{\@dmp{RAM=#1}\if@sites%
93 \PackageWarning{Do not use the RAM key in the presence of sites}\else%
94 \pdata@def{all}{intended}{RAM}{#1}\fi}
```

similarly, the `PI` keys are registered in `\prop@gen@PIs`.

```
95 \define@key{prop@gen}{PI}{\@dmp{PI=#1}%
96 \@ifundefined{prop@gen@PIs}{\xdef\prop@gen@PIs{#1}}{\xdef\prop@gen@PIs{\prop@gen@PIs,#1}}}
```

and the `pubspage` keys in `\prop@gen@pubspages`.

```
97 \define@key{prop@gen}{pubspage}{\@ifundefined{prop@gen@pubspages}%
98 {\xdef\prop@gen@pubspages{#1}}{\xdef\prop@gen@pubspages{\prop@gen@pubspages,#1}}}
```

the `importfrom` key reads the proposal data from its argument.

```
99 \define@key{prop@gen}{importfrom}{\message{importing proposal data from #1.pdata}\readpdata{#1}}
```

The rest of the keys just store their value.

```
100 \define@key{prop@gen}{instrument}{\def\prop@gen@instrument{#1}%
101 \pdata@def{prop}{gen}{instrument}{#1}\@dmp{inst=#1}}
102 \define@key{prop@gen}{title}{\def\prop@gen@title{#1}%
103 \pdata@def{prop}{gen}{title}{#1}}
104 \define@key{prop@gen}{acronym}{\gdef\prop@gen@acronym{#1}%
105 \pdata@def{prop}{gen}{acronym}{#1}\@dmp{acro=#1}}
106 \define@key{prop@gen}{acrolong}{\def\prop@gen@acrolong{#1}%
107 \pdata@def{prop}{gen}{acrolong}{#1}}
108 \define@key{prop@gen}{discipline}{\def\prop@gen@discipline{#1}%
109 \pdata@def{prop}{gen}{discipline}{#1}}
110 \define@key{prop@gen}{areas}{\def\prop@gen@areas{#1}%
111 \pdata@def{prop}{gen}{areas}{#1}}
112 \define@key{prop@gen}{start}{\def\prop@gen@start{#1}%
113 \pdata@def{prop}{gen}{start}{#1}}
114 \define@key{prop@gen}{months}{\def\prop@gen@months{#1}%
115 \pdata@def{prop}{gen}{months}{#1}}
116 \define@key{prop@gen}{since}{\def\prop@gen@since{#1}%
117 \pdata@def{prop}{gen}{since}{#1}}
118 \define@key{prop@gen}{totalduration}{\def\prop@gen@totalduration{#1}%
119 \pdata@def{prop}{gen}{totalduration}{#1}}
120 \define@key{prop@gen}{fundsuntil}{\def\prop@gen@fundsuntil{#1}%
121 \pdata@def{prop}{gen}{fundsuntil}{#1}}
```

and the default values, these will be used, if the author does not specify something better.

```
122 \newcommand\prop@gen@acro@default{ACRONYM}
123 \def\prop@gen@acro{\prop@gen@acro@default}
124 \newcommand\prop@gen@months@default{???months???}
125 \def\prop@gen@months{\prop@gen@months@default}
126 \newcommand\prop@gen@title@default{???Proposal Title???}
127 \def\prop@gen@title{\prop@gen@title@default}
```

```
128 \newcommand\prop@gen@instrument@default{??? Instrument ???}
129 \def\prop@gen@instrument{\prop@gen@instrument@default}
```

\prop@tl An auxiliary macro that is handy for making tables of WorkAddress data.

```
130 \newcommand\prop@tl[2]{\xdef\tab@line{}
131 \@for\tl@ext:={#1}\do{\xdef\tab@line{\tab@line&#2}}
132 \tab@line}
```

4.3 Proposal Appearance

We define the keys for the proposal appearance

```
133 \def\prop@gen@compactht{false}
134 \define@key{prop@gen}{compactht}[true]{\def\prop@gen@compactht{#1}}
135 ⟨/pdata⟩
```

emphbox

```
136 ⟨*cls⟩
137 \newmdenv[settings=\large]{emphbox}
```

4.4 Title Page

prop@proposal This internal environment is called in the `proposal` environment from the `proposal` class. The implementation here is only a stub to be substituted in a specialized class.

```
138 \newenvironment{prop@proposal}
139 {\thispagestyle{empty}%
140 \begin{center}
141   {\LARGE \prop@gen@instrument}\\[.2cm]
142   {\LARGE\textbf{\prop@gen@title}}\\[.3cm]
143   {\LARGE Acronym: {\prop@gen@acronym}}\\[.2cm]
144   {\large\today}\\[1em]
145   \begin{tabular}{c*{\the@PIs}{c}}
146     \prop@tl\prop@gen@PIs{\wa@ref{person}\tl@ext{name}}\\
147     \prop@tl\prop@gen@PIs{\wa@ref{institution}{\wa@ref{person}\tl@ext{affiliation}}{name}}
148   \end{tabular}\\[2cm]
149 \end{center}
150 \setcounter{tocdepth}{2}\tableofcontents\newpage\setcounter{page}{1}}
```

Now we come to the end of the environment:

```
151 {\section{List of Attachments}
152 \begin{itemize}
153 \@for\@I:=\prop@gen@PIs\do{%
154 \item Curriculum Vitae and list of publications for
155   \wa@ref{person}\@I{personaltitle} \wa@ref{person}\@I{name}
156 \end{itemize}}\newpage
157 \printbibliography[heading=warnpubs]}
```

proposal The `proposal` environment reads the metadata keys defined above, and if there were no `site` keys, then it defines keys RM and RAM (unless the `noRAM` package option was given) for the `workpackage` environment. Also it reads the project data file and opens up the project data file `\pdata@out`, which it also closes at the end.

The environment calls an internal version of the environment `prop@proposal` that can be customized by the specializing classes.

```
158 \newenvironment{proposal}[1][]{\readpdata\jobname
159 \ofpagetrue\setkeys{prop@gen}{#1}
160 \pdata@open\jobname
161 \if@sites\else
162 \define@key{workpackage}{RM}{\pdata@def{wp}\wp@id{RM}{##1}\@dmp{RM=##1}}
```

```
163 \if@RAM\define@key{workpackage}{RAM}{\pdata@def{wp}\wp@id{RAM}{##1}\@dmp{RAM=##1}}\fi
164 \fi
165 \newcounter{@PIs}
166 \@ifundefined{prop@gen@PIs}{}{\@for\@I:=\prop@gen@PIs\do{\stepcounter{@PIs}}}
167 \newcounter{@sites}
168 \@ifundefined{prop@gen@sites}{}{\@for\@I:=\prop@gen@sites\do{\stepcounter{@sites}}}
169 \setcounter{page}{0}
170 \begin{prop@proposal}}
```

Now we come to the end of the environment, we take care of the last page and print the references.

```
171 {\end{prop@proposal}
172 \pdata@def{prop}{page}{last}{\thepage}\ofpagefalse
173 \pdata@close}
174 ⟨/cls⟩
```

The `report` environment is similar, but somewhat simpler

`report`
```
175 ⟨∗reporting⟩
176 \newif\if@report\@reportfalse
177 \newenvironment{report}[1][]%
178 {\@reporttrue\readpdata\jobname%
179 \ofpagetrue\setkeys{prop@gen}{#1}%
180 \pdata@open\jobname%
181 \@ifundefined{prop@gen@PIs}{}{\newcounter{@PIs}\@for\@I:=\prop@gen@PIs\do{\stepcounter{@PIs}}}%
182 \@ifundefined{prop@gen@sites}{}{\newcounter{@sites}\@for\@I:=\prop@gen@sites\do{\stepcounter{@sites}}}%
183 \setcounter{page}{0}%
184 \begin{prop@report}}
185 {\end{prop@report}%
186 \pdata@def{prop}{page}{last}{\thepage}\ofpagefalse\newpage
187 \printbibliography[heading=warnpubs]
188 \pdata@close}
```

`prop@report`
```
189 \newenvironment{prop@report}
190 {\begin{center}
191   {\LARGE Final Project Report}\\[.2cm]
192   {\LARGE\textbf{\prop@gen@title}}\\[.3cm]
193   {\LARGE Acronym: {\prop@gen@acronym}}\\[.2cm]
194   {\large\today}\\[1em]
195   \begin{tabular}{c*{\the@PIs}{c}}
196     \prop@tl\prop@gen@PIs{\wa@ref{person}\tl@ext{name}}\\
197     \prop@tl\prop@gen@PIs{\wa@ref{institution}{\wa@ref{person}\tl@ext{affiliation}}{name}}
198 \end{tabular}\\[2cm]
199 \end{center}
200 \setcounter{tocdepth}{2}\tableofcontents\newpage\setcounter{page}{1}}
201 {}
202 ⟨/reporting⟩
```

`\site*`
```
203 ⟨∗cls⟩
204 \newcommand\site[1]{\hyperlink{site@#1@target}{\wa@ref{institution}{#1}{acronym}}}
205 \newcommand\sitename[1]{\hyperlink{site@#1@target}{\wa@ref{institution}{#1}{name}}}
```

4.5 Objectives

We first define a presentation macro for objectives

206 `\newcommand\objective@label[1]{O#1}`

We define the keys for the objectives environment

207 `\define@key{obj}{id}{\def\obj@id{#1}\@dmp{id=#1}}`
208 `\define@key{obj}{title}{\def\obj@title{#1}\@dmp{title=#1}}`
209 `\define@key{obj}{short}{\def\obj@short{#1}\@dmp{short=#1}}`

And a counter for numbering objectives

210 `\newcounter{objective}`

211 `\newenvironment{objective}[1][]`
212 `{\let\obj@id\relax\let\obj@title\relax\let\obj@short\relax%`
213 `\setkeys{obj}{#1}\stepcounter{objective}%`
214 `\goodbreak\smallskip\par\noindent%`
215 `\textbf{\objective@label{\arabic{objective}}:%`
216 `~\pdata@target{obj}{\obj@id}{\pdataref{obj}{\obj@id}{title}}\ignorespaces}%`
217 `\pdata@def{obj}\obj@id{label}{\objective@label\theobjective}%`
218 `\@ifundefined{obj@title}{}{\pdata@def{obj}\obj@id{title}\obj@title}%`
219 `\@ifundefined{obj@short}{}{\pdata@def{obj}\obj@id{short}\obj@short}}`
220 `{}`

221 `\newcommand\OBJref[1]{\pdataRef{obj}{#1}{label}}`
222 `\newcommand\OBJtref[1]{\pdataRef{obj}{#1}{label}: \pdataRef{obj}{#1}{title}}`

4.6 Work Packages and Work Groups

We first define keys for work groups (if we are in an IP).

223 `\ifwork@areas`
224 `\define@key{workarea}{id}{\def\wa@id{#1}\@dmp{id=#1}}`
225 `\define@key{workarea}{title}{\pdata@def{wa}\wa@id{title}{#1}}`
226 `\define@key{workarea}{short}{\pdata@def{wa}\wa@id{short}{#1}}`
227 `\define@key{workarea}{lead}{\pdata@def{wa}\wa@id{lead}{#1}}`
228 `\fi`

work packages have similar ones.

229 `\define@key{workpackage}{id}{\def\wp@id{#1}\@dmp{id=#1}}`
230 `\define@key{workpackage}{title}{\pdata@def{wp}\wp@id{title}{#1}}`
231 `\define@key{workpackage}{lead}{\pdata@def{wp}\wp@id{lead}{#1}\def\wp@lead{#1}\@dmp{lead=#1}}`
232 `\define@key{workpackage}{short}{\pdata@def{wp}\wp@id{short}{#1}}`
233 `\define@key{workpackage}{type}{\def\wp@type{#1}\pdata@def{wp}\wp@id{type}{#1}}`
234 `\define@key{workpackage}{wphases}{\def\wp@wphases{#1}\pdata@def{wp}\wp@id{wphases}{#1}}`

We define the constructors for the work package and work group labels and titles.

235 `\newcommand\wp@mk@title[1]{Work Package {#1}}`
236 `\newcommand\wp@label[1]{WP{#1}}`
237 `\ifwork@areas`
238 `\newcommand\wa@label[1]{WA{#1}}`
239 `\newcommand\wa@mk@title[1]{Work Area {#1}}`
240 `\fi`

The `wa` and `wp` counters are for the work packages and work groups, the counter `deliv` for deliverables.

241 `\ifwork@areas\newcounter{wa}\newcounter{wp}[wa]\else\newcounter{wp}\fi`
242 `\ifdelivs\newcounter{deliv}[wp]\fi`
243 `\newcounter{allwp}`

\update@*	update the list \@wps of the work packages in the local group and the list \@was work groups for the staff efforts table: if \@wps is undefined, then initialize the comma-separated list, otherwise
EdN:3	extend it.[3]

```
244 \newcommand\update@wps[1]{\@ifundefined{@wps}{\xdef\@wps{#1}}{\xdef\@wps{\@wps,#1}}}
245 \newcommand\update@tasks[1]{\@ifundefined{@tasks}{\xdef\@tasks{#1}}{\xdef\@tasks{\@tasks,#1}}}
246 \newcommand\update@deps[1]{\@ifundefined{task@deps}{\xdef\task@deps{#1}}{\xdef\task@deps{\task@deps,#1}}}
247 \ifwork@areas\def\update@was#1{\@ifundefined{@was}{\xdef\@was{#1}}{\xdef\@was{\@was,#1}}}\fi
```

\decode@wphase \decode@wphase decodes a string of the form ⟨start⟩-⟨end⟩!⟨force⟩ and defines the macros \wphase@start, \wphase@end, and \wphase@force with the three parts and also computes \wphase@len. The intermediate parsing macro \decode@p@start parses out the start (a number), and passes on to \decode@p@end, which parses out the end (another number) and the force string, which is either empty (if the !⟨force⟩ part is omitted) or of the form !⟨force⟩. In the first case the default value 1 is returned for \decode@force in the second ⟨force⟩.

```
248 \newcommand\decode@wphase[1]{\expandafter\decode@p@start#1@%
249 \local@count\wphase@end\advance\local@count by -\wphase@start%
250 \def\wphase@len{\the\local@count}}
251 \def\decode@p@start#1-#2@{\def\wphase@start{#1}\decode@p@end#2!@}
252 \def\decode@p@end#1!#2@{\def\wphase@end{#1}\def\@test{#2}%
253 \ifx\@test\@empty\def\wphase@force{1}\else\decode@p@force#2\fi}
254 \def\decode@p@force#1!{\def\wphase@force{#1}}
```

\startend@wphases We first iteratively decode the work phases, so that the last definition of \wphase@end remains, then we parse out the start of the first workphase to define \wphase@start

```
255 \def\wphases@start#1-#2@{\def\wphase@start{#1}}
256 \newcommand\startend@wphases[1]{\def\@test{#1}
257 \ifx\@test\@empty\def\wphase@start{0}\def\wphase@end{0}\else%
258 \@for\@I:=#1\do{\expandafter\decode@p@start\@I @}
259 \expandafter\wphases@start#1@\fi}
```

with these it is now relatively simple to define the interface macros.

work@package The workpackage environment collects the keywords, steps the counters, writes the metadata to the aux file, updates the work packages in the local group, generates the work package number \wp@num.

```
260 \newcounter{wp@RM}
261 \if@RAM\newcounter{wp@RAM}\fi
262 \newenvironment{work@package}[1][]%
263 {\def\wp@wphases{0-0}% default values
264 \setkeys{workpackage}{#1}\stepcounter{wp}\stepcounter{allwp}%
265 \startend@wphases\wp@wphases%
266 \pdata@def{wp}\wp@id{start}\wphase@start\pdata@def{wp}\wp@id{end}\wphase@end%
267 \@ifundefined{wp@type}{}{\pdata@def{wp}\wp@id{type}\wp@type}%
268 \let\@tasks=\relax%
269 \edef\wp@num{\ifwork@areas\thewa.\fi\thewp}%
270 \pdata@def{wp}\wp@id{label}{\wp@label\wp@num}%
271 \pdata@def{wp}\wp@id{number}{\thewp}%
272 \pdata@def{wp}\wp@id{page}{\thepage}%
273 \update@wps\wp@id%
274 \edef\wp@num{\ifwork@areas\thewa.\fi\thewp}%
275 \pdata@def{wp}{\wp@id}{num}{\thewp}%
```

If we have sites, we have to compute the total RM and RAM for this WP.

```
276 \if@sites%
277 \setcounter{wp@RM}{0}\if@RAM\setcounter{wp@RAM}{0}\fi%
```

---

[3]EDNOTE: with the current architecture, we cannot have work areas that do not contain work packages, this leads to the error that wps is undefined in endworkplan

```
278 \@for\@site:=\prop@gen@sites\do{%
279 \edef\@RM{\pdataref@num\wp@id\@site{RM}}\addtocounter{wp@RM}{\@RM}%
280 \if@RAM\edef\@RAM{\pdataref@num\wp@id\@site{RAM}}\addtocounter{wp@RAM}{\@RAM}\fi}
281 \pdata@def{wp}\wp@id{RM}{\thewp@RM}%
282 \if@RAM\pdata@def{wp}\wp@id{RAM}{\thewp@RAM}\fi%
283 \fi}% if@sites
284 {\@ifundefined{@tasks}{}{\pdata@def{\wp@id}{task}{ids}\@tasks}}
```

workpackage   With this, it becomes simple to define a work package environment. We consider two cases, if
we have sites, then we make a header table. If not, we can make things much simpler: we just
generate a subsection

```
285 \newenvironment{workpackage}[1][]%
286 {\begin{work@package}[#1]%
287 %\if@wpsubsection\subsubsection*{{\wp@mk@title\thewp}: \pdataref{wp}\wp@id{title}}\fi
288 \if@sites\goodbreak\medskip\wpheadertable%
289 \else\subsubsection*{{\wptitle} (\wprm)}\fi%
290 \addcontentsline{toc}{paragraph}{{\wp@mk@title\thewp}: \pdataref{wp}\wp@id{title}}%
291 \noindent\ignorespaces}
292 {\end{work@package}}
```

EdN:4 \wptitle   [4]

```
293 \newcommand\wptitle{\wp@mk@title{\wp@num}: \pdata@target{wp}{\wp@id}{\pdataref{wp}\wp@id{title}}}
```

EdN:5 \wprm   [5]

```
294 \newcommand\wprm{\pdataref@safe{wp}\wp@id{RM}\if@RAM\ RM+\pdataref{wp}\wp@id{RAM} RAM\fi}
```

@site@contributes   Called as \if@site@contributes{⟨site⟩}{⟨tokens⟩} the following happens: If \prop@gen@compactht
is \@true (set by the compactht attribute on the proposal environment), then ⟨tokens⟩ is pro-
cessed. Otherwise, ⟨tokens⟩ is only processed if ⟨site⟩ contributes to the current work package (i.e
the RM $\neq 0$ and RAM $\neq 0$)

```
295 \newcount\site@contribution%
296 \newcommand\if@site@contributes[2]{%
297 \ifx\prop@gen@compactht\@true
298 \if@RAM\ifnum\pdataref@num\wp@id{#1}{RM} > 0 \ifnum \pdataref@num\wp@id{#1}{RAM} > 0 #2\fi\fi
299 \else\ifnum\pdataref@num\wp@id{#1}{RM} > 0 #2\fi\fi
300 \else #2\fi}
```

\wp@sites@line   The following macro computes the sites line (in the token register \wp@sites@line), the efforts
\wp@efforts@line line (in \wp@efforts@line), and the sites number (in the counter \sites@num) for later inclusion
\wp@sites@num in the \wpheadertable. If \prop@gen@compactht is \@true, then no sites without contributions
are listed in the table.

```
301 \newcounter{wp@sites@num}
302 \newcommand\wp@sites@efforts@lines{%
303 \setcounter{wp@sites@num}{0}
304 {\let\G@refundefinedtrue=\relax\let\@latex@warning=\relax%
305 \let\site\relax\let\textbf\relax\let\sum@style\relax\let\lead@style\relax%
306 \let\pn\relax\let\sys\relax%
307 \xdef\wp@sites@line{\wp@legend@site}\xdef\wp@efforts@line{\wp@legend@effort}%initialize lines
308 \@for\@site:=\prop@gen@sites\do{\if@site@contributes\@site{\stepcounter{wp@sites@num}}%
309 \xdef\wp@sites@line{\wp@sites@line%
310 \if@site@contributes\@site{&\ifx\@site\wp@lead\lead@style{\site{\@site}}\else\site{\@site}\fi}}%
311 \xdef\wp@efforts@line{\wp@efforts@line%
312 \if@site@contributes\@site{&\pdataref@safe\wp@id\@site{RM}\if@RAM+\pdataref@safe\wp@id\@site{RAM}\fi}}%
313 \xdef\wp@sites@line{\wp@sites@line&\sum@style{\wp@legend@all}}%
314 \xdef\wp@efforts@line{\wp@efforts@line&
315 \sum@style{\textbf{\pdataref{wp}\wp@id{RM}\if@RAM+\pdataref{wp}\wp@id{RAM}\fi}}}}}
```

---

[4]EdNote: document above
[5]EdNote: document above

16

\wpheadertable This macro computes the default work package header table, if there are sites.

```
316 \newcommand\wpheadertable{%
317 \wp@sites@efforts@lines%
318 \par\noindent\begin{tabular}{|l||l|*{\thewp@sites@num}{c|}|c|}\hline%
319 \textbf{\wp@mk@title{\wp@num}}&\wp@sites@line\\\hline%
320 \textsf{\pdata@target{wp}{\wp@id}{\pdataref{wp}\wp@id{title}}} &\wp@efforts@line\\\hline%
321 \end{tabular}\smallskip\par\noindent\ignorespaces}
```

and now multilinguality support

```
322 \newcommand\wp@legend@site{Site}
323 \newcommand\wp@legend@effort{Effort\if@RAM{ (RM+RAM)}\fi}
324 \newcommand\wp@legend@all{\textbf{all}}
```

workarea the `workarea` environment for work groups is almost the same, but we also have to initialize the work package counters. Also, the efforts can be computed from the work packages in this group via the `wa@effort` counter

```
325 \newcounter{prop@RM}\if@RAM\newcounter{prop@RAM}\fi
326 \ifwork@areas
327 \newcounter{wa@RM}\if@RAM\newcounter{wa@RAM}\fi\newcounter{wa@wps}
328 \newenvironment{workarea}[1][]
329 {\setkeys{workarea}{#1}
330 \let\@wps=\relax
331 \stepcounter{wa}
332 \pdata@def{wa}{\wa@id}{label}{\wa@label\thewa}
333 \pdata@def{wa}{\wa@id}{number}{\thewa}
334 \pdata@def{wa}{\wa@id}{page}{\thepage}
335 \update@was{\wa@id}
336 \pdata@def{wa}{\wa@id}{num}{\thewa}
337 \setcounter{wa@RM}{0}\if@RAM\setcounter{wa@RAM}{0}\fi\setcounter{wa@wps}{0}
338 \edef\@@wps{\pdataref@aux\wa@id{wp}{ids}}
339 \@for\@wp:=\@@wps\do{\stepcounter{wa@wps}}%
340 \if@sites
341 \@for\@site:=\prop@gen@sites\do{%
342   \edef\@RM{\pdataref@num\@wp\@site{RM}}
343   \if@RAM\edef\@RAM{\pdataref@num\@wp\@site{RAM}}\fi
344   \addtocounter{wa@RM}{\@RM}\addtocounter{prop@RM}{\@RM}
345   \if@RAM\addtocounter{wa@RAM}{\@RAM}\addtocounter{prop@RAM}{\@RAM}\fi
346 \else
347 \edef\@RM{\pdataref@num{wp}\@wp{RM}}
348 \if@RAM\edef\@RAM{\pdataref@num{wp}\@wp{RAM}}\fi
349 \addtocounter{wa@RM}{\@RM}\addtocounter{prop@RM}{\@RM}
350 \if@RAM\addtocounter{wa@RAM}{\@RAM}\addtocounter{prop@RAM}{\@RAM}\fi
351 \fi}
352 \pdata@def{wa}\wa@id{RM}\thewa@RM
353 \pdata@def{prop}{all}{RM}\theprop@RM
354 \if@RAM
355 \pdata@def{wa}\wa@id{RAM}\thewa@RAM
356 \pdata@def{prop}{all}{RAM}\theprop@RAM
357 \fi
358 \subsubsection*{{\wa@mk@title\thewa}: {\pdata@target{wa}\wa@id{\pdataref{wa}\wa@id{title}}}}
359 \addcontentsline{toc}{subsubsection}{{\wa@mk@title\thewa}: \pdataref{wa}\wa@id{title}}%
360 \ignorespaces}
361 {\@ifundefined{@wps}{}{\pdata@def\wa@id{wp}{ids}\@wps}\pdata@def\wa@id{wp}{count}\thewa@wps}\fi
```

workplan The `workplan` environment sets up the accumulator macros `\@wps`, `\@was`, for the collecting the identifiers of work packages and work groups. At the end of the workplan description it writes out their content to the aux file for reference.

```
362 \ifdelivs\newwrite\wpg@delivs\fi
```

```
363 \newenvironment{workplan}%
364 {\ifdelivs\immediate\openout\wpg@delivs=\jobname.delivs\fi
365 \ifwork@areas\let\@was=\relax\else\let\@wps=\relax\fi}%
366 {\@ifundefined{task@deps}{}{\pdata@def{all}{task}{deps}{\task@deps}}
367 \pdata@def{all}{task}{count}{\thealltasks}
368 \ifwork@areas
369 \@ifundefined{@was}{}{\pdata@def{all}{wa}{ids}\@was}
370 \else
371 \@ifundefined{@wps}{}{\pdata@def{all}{wp}{ids}\@wps}
372 \fi
373 \ifdelivs\@ifundefined{mile@stones}{}
374 {\@for\@I:=\mile@stones\do{%
375 \pdata@def{mile}\@I{delivs}{\@ifundefined{\@I delivs}{}{\csname\@I delivs\endcsname}}}}\fi
376 \ifwork@areas\pdata@def{all}{wa}{count}{\thewa}\fi
377 \pdata@def{all}{wp}{count}{\theallwp}
378 \ifdelivs
379 \pdata@def{all}{deliverables}{count}{\thedeliverable}
380 \pdata@def{all}{milestones}{count}{\themilestone}
381 \fi
382 \ifdelivs\closeout\wpg@delivs\fi}
```

4.7 Milestones and Deliverables

deliv@error this macro raises an error if deliverable commands are used without the `deliverables` option being set.

```
383 \newcommand\deliv@error{\PackageError{proposal}
384 {To use use deliverables, you have to specify the option 'deliverables'}}
```

wpdelivs

```
385 \newenvironment{wpdelivs}{\begin{wp@delivs}}{\end{wp@delivs}}
```

wp@delivs

```
386 \newenvironment{wp@delivs}
387 {\ifdelivs\textbf\deliv@legend@delivs:\\[-3ex]%
388 \begin{compactdesc}\else\deliv@error\fi}
389 {\ifdelivs\end{compactdesc}\fi}
```

and now multilinguality support

```
390 \newcommand\deliv@legend@delivs{Deliverables}
```

\wadelivs

```
391 \newenvironment{wadelivs}
392 {\textbf\deliv@legend@delivs:\\[-3ex]\begin{wp@delivs}}
393 {\end{wp@delivs}}
```

\lec This macro is generally useful to put a comment at the end of the line, possibly making a new one if there is not enough space.

```
394 \newcommand\lec[1]{\strut\hfil\strut\null\nobreak\hfill\hbox{$\leadsto$#1}\par}
```

\deliv@label

```
395 \newcommand\deliv@label[1]{M{#1}}
```

\delivref This macro is generally useful to put a comment at the end of the line, possibly making a new one if there is not enough space.

```
396 \newcommand\delivref[2]{\pdataRef{deliv}{#1#2}{label}}
397 \newcommand\delivtref[2]{\pdataRef{deliv}{#1#2}{label}: \pdataRef{deliv}{#1#2}{short}}
```

\wpg@deliv   We first define the keys

```
398 \define@key{deliv}{id}{\def\deliv@id{#1}}
399 \define@key{deliv}{due}{\def\deliv@due{#1}}
400 \define@key{deliv}{dissem}{\def\deliv@dissem{#1}}
401 \define@key{deliv}{nature}{\def\deliv@nature{#1}}
402 \define@key{deliv}{miles}{\def\deliv@miles{#1}}
403 \define@key{deliv}{short}{\def\deliv@short{#1}}
```

The \wpdeliv macro cycles over the due dates and generates the relevant entries into the deliverables file. The first step is to write the general metadata to the pdata file.

```
404 \newcounter{deliverable}
405 \newcommand{\wpg@deliv}[3]{% keys, title, type
406 \stepcounter{deliverable}
407 \let\deliv@miles=\relax% clean state
408 \def\@type{#3}\def\@wp{wp}% set up ifx
409 \def\wpg@id{\csname #3@id\endcsname}
410 \setkeys{deliv}{#1}\stepcounter{deliv}% set state
411 \ifx\@type\@wp\def\current@label{\deliv@label{\ifwork@areas\thewa.\fi\thewp.\thedeliv}}
412 \else\def\current@label{\deliv@label{\thewa.\thedeliv}}\fi
413 \pdata@def{deliv}{\wpg@id\deliv@id}{label}{\current@label}
414 \pdata@def{deliv}{\wpg@id\deliv@id}{title}{#2}
415 \@ifundefined{deliv@short}
416 {\pdata@def{deliv}{\wpg@id\deliv@id}{short}{#2}}
417 {\pdata@def{deliv}{\wpg@id\deliv@id}{short}{\deliv@short}}
418 \pdata@def{deliv}{\wpg@id\deliv@id}{nature}{\deliv@nature}
419 \pdata@def{deliv}{\wpg@id\deliv@id}{dissem}{\deliv@dissem}
```

Then we iterate over the due dates and generate an entry for teach of them.

```
420 \@ifundefined{deliv@due}{}{%
421 \@for\@I:=\deliv@due\do{\protected@write\wpg@delivs{}{\string\deliverable%
422 {\ifnum\@I<10 0\@I\else\@I\fi}% sort key
423 {\@I}% due date
424 {\current@label}% label
425 {\@ifundefined{deliv@id}{\protect\G@refundefinedtrue\@latex@warning{key 'id' for Deliv #1
426       undefined}??}{\wpg@id\deliv@id}}% id
427 {\@ifundefined{deliv@dissem}{\protect\G@refundefinedtrue\@latex@warning{key 'dissem' for
428       Deliv #1 undefined}??}{\deliv@dissem}}% dissemination level
429 {\@ifundefined{deliv@nature}{\protect\G@refundefinedtrue\@latex@warning{key 'nature' for Deliv
430       #1 undefined}??}{\deliv@nature}}% nature
431 {#2}
432 {\ifx\@type\@wp{WP\ifwork@areas\thewa.\fi\thewp}\else{WA\thewa}\fi}}}}%WP
```

And finally, we generate the entry into the deliverables table.

```
433 \item[\current@label: (Month \deliv@due; nature: \deliv@nature, dissem.: \deliv@dissem)] \pdata@target{deliv}
434 \@ifundefined{deliv@miles}{}{% print the milestones and update their deliverables
435 \let\m@sep=\relax% do not print the separator the first time round
436 \lec{\@for\@I:=\deliv@miles\do{% Iterate over the milestones mentioned
437 \m@sep\pdataRef{mile}{\@I}{label}% print the milestone reference
438 \let\m@sep=,}}%set the separator for the next times
439 \def\d@sep{,}
440 \@for\@I:=\deliv@miles\do{% Iterate over the milestones mentioned
441 \expandafter\ifx\csname\@I delivs\endcsname\relax% Check that the miles@delivs is empty
442  {\expandafter\xdef\csname\@I delivs\endcsname{\wpg@id\deliv@id}}% if so, skip the separator
443   \else\expandafter\xdef\csname\@I delivs\endcsname%if not add it
444      {\csname\@I delivs\endcsname\d@sep\wpg@id\deliv@id}\fi}}}
```

Now, we only need to instantiate

wadeliv

```
445 \newenvironment{wadeliv}[2][]{\ifdelivs\wpg@deliv{#1}{#2}{wa}\else\deliv@error\fi}{}
```

```
446 \newenvironment{wpdeliv}[2][]{\ifdelivs\wpg@deliv{#1}{#2}{wp}\else\deliv@error\fi}{}
```

\milestone@label

```
447 \newcommand\milestone@label[1]{M{#1}}
```

\mileref   This macro is generally useful to put a comment at the end of the line, possibly making a new one if there is not enough space.

```
448 \newcommand\mileref[1]{\pdataRef{mile}{#1}{label}}
449 \newcommand\miletref[1]{\pdataRef{mile}{#1}{label}: \pdataRef{mile}{#1}{short}}
```

\milestone   create a new milestone, initialize its deliverables accumulator macro, set up hyperlinking, and extend the milestones list.

```
450 \newcounter{milestone}
451 \define@key{milestone}{id}{\gdef\mile@id{#1}}
452 \define@key{milestone}{month}{\gdef\mile@month{#1}}
453 \define@key{milestone}{verif}{\gdef\mile@verif{#1}}
454 \newcommand\milestone[3][]{%
455 \ifdelivs%
456 \setkeys{milestone}{#1}\stepcounter{milestone}%
457 \pdata@def{mile}\mile@id{label}{\milestone@label{\themilestone}}%
458 \pdata@def{mile}\mile@id{month}{\mile@month}%
459 \pdata@def{mile}\mile@id{verif}{\mile@verif}%
460 \pdata@def{mile}\mile@id{title}{#2}%
461 \@ifundefined{mile@stones}{\xdef\mile@stones{\mile@id}}{\xdef\mile@stones{\mile@stones,\mile@id}}%
462 \@milestone{#1}{#2}{#3}% presentation
463 \else\deliv@error\fi}
```

\@milestone   the corresponding presentation macro.

```
464 \newcommand\@milestone[3]{%
465 \pdata@target{mile}\mile@id{\textbf{\milestone@label\themilestone}}&
466 \textbf{#2} &
467 \prop@milesfor\mile@id &
468 \pdataref{mile}\mile@id{month} &
469 \pdataref{mile}\mile@id{verif}\\\hline
470 \multicolumn{5}{|p{14cm}|}{#3}\\\hline\hline}
```

```
471 \newenvironment{milestones}{\begin{@milestones}}{\end{@milestones}}
```

```
472 \newenvironment{@milestones}
473 {\ifdelivs\begin{longtable}{|l|p{4cm}|p{5cm}|l|p{2.5cm}|}\hline
474 \#&\miles@legend@name&\miles@legend@involved&\miles@legend@month&\miles@legend@verif\\\hline\hline%
475 \else\deliv@error\fi}
476 {\ifdelivs\end{longtable}%
477 \footnotetext\miles@legend@footnote\fi}
```

now the multilinguality support

```
478 \newcommand\miles@legend@name{Name}
479 \newcommand\miles@legend@month{Mo}
480 \newcommand\miles@legend@verif{Means of Verif.}
481 \newcommand\miles@legend@involved{WPs\footnotemark/Deliverables involved}
482 \newcommand\miles@legend@footnote{The work package number is the first number in the deliverable number.}
```

\prop@milesfor   the due date is the first argument to facilitate sorting

```
483 \newcommand\prop@milesfor[1]{\edef\@delivs{\pdataref@safe{mile}{#1}{delivs}}%
484 \let\m@sep=\relax\def\new@sep{,\ }%
485 \@for\@I:=\@delivs\do{\m@sep\pdataRef{deliv}\@I{label}\let\m@sep=\new@sep}}
```

\deliverable   the first argument is an extended due date to facilitate sorting.

486 \newcommand{\deliverable}[8]{\pdataRef{deliv}{#4}{label}&#7&#8&#6&#5&#2\\\hline}%sortkey,due,label,id,title,t

deliverables

487 \newenvironment{deliverables}[1]{\ifdelivs\begin{longtable}{|l|p{#1}|l|l|l|l|}\hline
488 \#&\delivs@legend@name&\delivs@legend@wp&\delivs@legend@nature&
489 \delivs@legend@level&\delivs@legend@due\\\hline\hline\else\deliv@error\fi}
490 {\ifdelivs\end{longtable}\fi}

 now the multilingual support

491 \newcommand\delivs@legend@name{Deliverable name}
492 \newcommand\delivs@legend@wp{WP}
493 \newcommand\delivs@legend@nature{Nature}
494 \newcommand\delivs@legend@level{Level}
495 \newcommand\delivs@legend@due{Due}

\inputdelivs

496 \newcommand{\inputdelivs}[1]{%
497 \begin{deliverables}{#1}%
498 \IfFileExists{\jobname.deliverables}%
499 {\input{\jobname.deliverables}}%
500 {\IfFileExists{\jobname.delivs}{\input{\jobname.delivs}}{}}
501 \end{deliverables}}

4.8   Tasks and Work Phases

tasklist

502 \newenvironment{tasklist}
503 {\begin{compactenum}}{\end{compactenum}}

 The next step is to

504 \newcommand\task@label[1]{T#1}

 We define the keys for the task macro

505 \define@key{task}{id}{\def\task@id{#1}\@dmp{id=#1}}
506 \define@key{task}{wphases}{\def\task@wphases{#1}\pdata@def{task}{\taskin\task@id\wp@id}{wphases}{#1}\@dmp{wph
507 \define@key{task}{requires}{\@requires\task@id{#1}\@dmp{req=#1}}
508 \define@key{task}{title}{\def\task@title{#1}\pdata@def{task}{\taskin\task@id\wp@id}{title}{#1}}
509 \define@key{task}{lead}{\def\task@lead{#1}\pdata@def{task}{\taskin\task@id\wp@id}{lead}{#1}\@dmp{lead=#1}}
510 \define@key{task}{partners}{\def\task@partners{#1}\pdata@def{task}{\taskin\task@id\wp@id}{partners}{#1}\@dmp{
511 \define@key{task}{PM}{\def\task@PM{#1}\pdata@def{task}{\taskin\task@id\wp@id}{PM}{#1}\@dmp{PM=#1}}

 then we define an auxiliary function that gives them sensible defaults and sets the internal macros.

512 \def\task@set#1{\edef\task@id{task\thetask@all}
513 \def\task@wphases{0-0}\def\task@partners{}\def\task@lead{}
514 \setkeys{task}{#1}}

@post@title@space   make the space after the title tweakable

515 \def\task@post@title@space{\quad}

task

516 \newcounter{alltasks}
517 \def\task@post@title@space{\quad}
518 \newenvironment{task}[1][]%
519 {\stepcounter{alltasks}
520 \@task{#1}\item[\pdata@target{task}{\taskin\task@id\wp@id}{\task@label{\thetask@wp}}]%
521 \@ifundefined{task@title}{}{\textbf\task@title}\task@post@title@space%
522 \def\@initial{0-0}\ifx\task@wphases\@initial\else%

523 `\ (\let\@@sep=\relax\@for\@I:=\task@wphases%`
524 `\do{\decode@wphase\@I\@@sep\show@wphase\wphase@start\wphase@end\wphase@force\let\@@sep=\sep@wphases}%`
525 `\ifx\task@lead\@empty\else; \task@legend@partners: \site\task@lead~(\legend@lead)\fi%`
526 `\ifx\task@partners\@empty\else\@for \@I:=\task@partners\do{, \site\@I}\fi)\\\fi}`
527 `{\ignorespaces}`

now the multilingual support and presentation configuration

528 `\newcommand\month@label[1]{M#1}`
529 `\newcommand\show@wphase[3]{\def\@test{#3}\month@label{#1}-\month@label{#2}%`
530 `\ifx\@test\@empty\@ #3}`
531 `\newcommand\sep@wphases{; }`
532 `\newcommand\legend@partners{Partners}`
533 `\newcommand\legend@lead{lead}`
534 `\newcommand\task@label@long{Task}`

\@task The `\@task` macro is a internal macro which takes a bunch of keyword keys and writes their values to the aux file.

535 `\newcounter{task@all}\newcounter{task@wp}[wp]`
536 `\newcount\task@@end`
537 `\def\@task#1{\stepcounter{task@all}\stepcounter{task@wp}%`
538 `\task@set{#1}%`
539 `\pdata@def{task}{\taskin\task@id\wp@id}{wphases}\task@wphases`
540 `\pdata@def{task}{\taskin\task@id\wp@id}{label}{\task@label\thetask@wp}%`
541 `\pdata@def{task}{\taskin\task@id\wp@id}{number}{\thetask@wp}%`
542 `\pdata@def{task}{\taskin\task@id\wp@id}{page}{\thepage}%`
543 `\update@tasks{\taskin\task@id\wp@id}}`

\workphase

544 `\newcommand\workphase[1]{\PackageError{proposal}`
545 `  {The \protect\workphase macro is deprecated,\MessageBreak`
546 `    use the attributes wphase on the workpackage environment instead!}}`

\localtaskref

547 `\newcommand\localtaskref[1]{\pdataRef{task}{\wp@id @#1}{label}}`

\taskref

548 `\newcommand\taskin[2]{#2@#1}`
549 `\newcommand\taskref[2]{\WPref{#1}.\pdataRef{task}{#1@#2}{label}}`
550 `\newcommand\taskreflong[2]{\WPref{#1}.\pdataRef{task}{#2}{label}}`
551 `\newcommand\tasktref[2]{\WPref{#1} (\task@label@long \pdataRef{task}{#1@#2}{number})}`
552 `\newcounter{gantt@deps}`
553 `\def\@requires#1#2{\stepcounter{gantt@deps}%`
554 `\edef\dep@id{taskdep\thegantt@deps}%`
555 `\pdata@def{taskdep}\dep@id{from}{\taskin{#1}\wp@id}%`
556 `\pdata@def{taskdep}\dep@id{to}{#2}%`
557 `\update@deps\dep@id}`
558 ⟨/cls⟩

4.9 Project Data, Referencing & Hyperlinking

\pdata@* `\pdata@out` is the file handle for the project data file, we define internal macros to open and close it.

559 ⟨*pdata⟩
560 `\newif\ifwork@areas\work@areastrue`
561 `\DeclareOption{noworkareas}{\work@areasfalse}`
562 `\ProcessOptions`
563 `\RequirePackage{xspace}`
564 `\newwrite\pdata@out`

`\newcommand\pdata@open[1]{\immediate\openout\pdata@out=#1.pdata}`
566 `\newcommand\pdata@close{\closeout\pdata@out}`

\readpdata This macro reads the project data file and its error handling

567 `\newcommand\readpdata[1]{\IfFileExists{#1.pdata}`
568 `{\message{proposal: Reading Project Data}\makeatletter\input{#1.pdata}\makeatother}`
569 `{proposal: No Project Data found, (forward) references may be compromised}}`

\pdata@target This internal macro makes a hyper-target: `\pdata@target{⟨cat⟩}{⟨id⟩}{⟨label⟩}` prints ⟨label⟩ with a target name ⟨cat⟩@⟨id⟩@target attached to it.

570 `\newcommand\pdata@target[3]{\hypertarget{#1@#2@target}{#3}}`

\pdata@def This macro writes an `\@pdata@def` command to the current aux file and also executes it.

571 `\newcommand\pdata@def[4]{%\@pdata@def{#1}{#2}{#3}{#4}%`
572 `  \protected@write\pdata@out{}{\string\@pdata@def{#1}{#2}{#3}{#4}}}`

\@pdata@def This macro stores the value of its last argument in a custom macro for reference.

573 `\newcommand\@pdata@def[4]{\expandafter\gdef\csname #1@#2@#3\endcsname{#4}}`

\pdataref

574 `\newcommand\pdataref[3]{\@ifundefined{#1@#2@#3}%`
575 `                {\protect\G@refundefinedtrue\@latex@warning{#3 for #1 #2 undefined}??}%`
576 `                 {\csname #1@#2@#3\endcsname}}%`
577 `\newcommand\pdataref@aux[3]{\@ifundefined{#1@#2@#3}{??}{\csname #1@#2@#3\endcsname}}%`
578 `\newcommand\pdataref@num[3]{\@ifundefined{#1@#2@#3}{0}{\csname #1@#2@#3\endcsname}}%`
579 `\newcommand\pdataref@safe[3]{\@ifundefined{#1@#2@#3}{}{\csname #1@#2@#3\endcsname}}%`

\pdataRef

580 `\newcommand\pdataRef[3]{\@ifundefined{#1@#2@#3}%`
581 `{\protect\G@refundefinedtrue\@latex@warning{#3 for #1 #2 undefined}??}%`
582 `{\hyperlink{#1@#2@target}{\csname #1@#2@#3\endcsname}}}`

\pdatacount

583 `\newcommand\prop@count[1]{\ifcase #1 zero\or one\or two\or three\or four\or five\or six\or seven \or`
584 `  eight\or nine\or ten\or eleven \or twelve\else#1\fi}`
585 `\newcommand\pdatacount[2]{\prop@count{\pdataref@num{#1}{#2}{count}}}`

\pn*

586 `\newcommand\pn{\pdataref{prop}{gen}{acronym}\xspace}`
587 `\newcommand\pnlong{\pdataref{prop}{gen}{acrolong}\xspace}`

\W*ref

588 `\newcommand\WPref[1]{\pdataRef{wp}{#1}{label}}`
589 `\newcommand\WPtref[1]{\pdataRef{wp}{#1}{label}: \pdataRef{wp}{#1}{short}}`
590 `\ifwork@areas`
591 `\newcommand\WAref[1]{\pdataRef{wa}{#1}{label}}`
592 `\newcommand\WAtref[1]{\pdataRef{wa}{#1}{label}: \pdataRef{wa}{#1}{title}}`
593 `\fi`
594 `⟨/pdata⟩`

4.10   The Work Package Table

\prop@lead

595 `⟨*cls⟩`
596 `\newcommand\prop@lead[1]{\@ifundefined{wp@#1@lead}%`
597 `{\protect\G@refundefinedtrue\@latex@warning{lead for WP #1 undefined}??}%`
598 `{\csname wp@#1@lead\endcsname}}`

```
599 \definecolorset{gray/rgb/hsb/cmyk}{}{}%
600 {leadgray,.90/.90,.90,.90/0,0,.90/0,0,0,.10;%
601 wagray,.70/.70,.70,.70/0,0,.70/0,0,0,.30}
602 \newcommand\sum@style[1]{\cellcolor{wagray}{\textbf{#1}}}
603 \newcommand\wa@style[1]{\cellcolor{wagray}{\textbf{#1}}}
604 \newcommand\wp@style[1]{#1}
605 \newcommand\lead@style[1]{\cellcolor{leadgray}{\textit{#1}}}
606 \newcommand\wp@lead@style@explained{light gray italicised}
```

wp@figure

```
607 \newcounter{wpfig@options}
608 \define@key{wpfig}{size}{\def\wpfig@size{#1}\@dmp{size=#1}}
609 \def\@true{true}
610 \def\wpfig@pages{false}
611 \define@key{wpfig}{pages}[true]{\def\wpfig@pages{#1}\stepcounter{wpfig@options}}
612 \def\wpfig@type{false}
613 \define@key{wpfig}{type}[true]{\def\wpfig@type{#1}\stepcounter{wpfig@options}}
614 \def\wpfig@start{false}
615 \define@key{wpfig}{start}[true]{\def\wpfig@start{#1}\stepcounter{wpfig@options}}
616 \def\wpfig@length{false}
617 \define@key{wpfig}{length}[true]{\def\wpfig@length{#1}\stepcounter{wpfig@options}}
618 \def\wpfig@end{false}
619 \define@key{wpfig}{end}[true]{\def\wpfig@end{#1}\stepcounter{wpfig@options}}
620 \def\@sw#1{\begin{sideways}#1\end{sideways}}
621 \newenvironment{wp@figure}{\begin{figure}[ht]\wpfig@style\begin{center}
622 {\let\@sw\relax\let\textbf\relax\let\site\relax\let\pn\relax\let\sys\relax%
623 \gdef\wpfig@headline{\wpfig@legend@wap&\wpfig@legend@title%
624 \ifx\wpfig@type\@true&\wpfig@legend@type\fi%
625 \ifx\wpfig@pages\@true&\@sw{\wpfig@legend@page}\fi%
626 \ifx\wpfig@start\@true&\@sw{\wpfig@legend@start}\fi%
627 \ifx\wpfig@length\@true&\@sw{\wpfig@legend@length}\fi
628 \ifx\wpfig@end\@true&\@sw{\wpfig@legend@end}\fi}%
629 \if@sites%
630 \@for\@site:=\prop@gen@sites\do{%
631 \xdef\wpfig@headline{\wpfig@headline&\@sw{\wpfig@legend@siteRM{\@site}}}%
632 \if@RAM\xdef\wpfig@headline{\wpfig@headline&\@sw{\wpfig@legend@siteRAM{\@site}}}\fi}%
633 \xdef\wpfig@headline{\wpfig@headline&\@sw{\wpfig@legend@totalRM}}%
634 \if@RAM\xdef\wpfig@headline{\wpfig@headline&\@sw{\wpfig@legend@totalRAM}}\fi%
635 \else% if@sites
636 \xdef\wpfig@headline{\wpfig@headline &\@sw{\wpfig@legend@RM}\if@RAM&\@sw{\wpfig@legend@RAM}\fi}
637 \fi}%if@sites
638 \if@RAM\begin{tabular}{|l|l|*{\thewpfig@options}{r|}*{\the@sites}{r|r|}|r|r|}\hline
639 \else\begin{tabular}{|l|l|*{\thewpfig@options}{r|}|*{\the@sites}{r|}|r|}\hline\fi%|
640 \wpfig@headline\\\hline\hline}
641 {\end{tabular}\smallskip\\
642 \wpfig@legend@RAM@expl
643 \if@sites; \wpfig@legend@lead@expl\fi
644 \caption{\wpfig@legend@caption}\label{fig:wplist}
645 \end{center}\end{figure}}}
```

and now multilinguality support

```
646 \newcommand\wpfig@legend@wap{\textbf{\ifwork@areas{WA/P}\else{WP}\fi}}
647 \newcommand\wpfig@legend@title{\textbf{Title}}
648 \newcommand\wpfig@legend@type{\textbf{type}}
649 \newcommand\wpfig@legend@page{\textbf{page}}
650 \newcommand\wpfig@legend@start{\textbf{start}}
```

```
651 \newcommand\wpfig@legend@length{\textbf{length}}
652 \newcommand\wpfig@legend@end{\textbf{end}}
653 \newcommand\wpfig@legend@siteRM[1]{\site{#1}\if@RAM\ RM\fi}
654 \newcommand\wpfig@legend@siteRAM[1]{\site{#1}\ RAM}
655 \newcommand\wpfig@legend@totalRM{total\if@RAM\ RM\fi}
656 \newcommand\wpfig@legend@totalRAM{total RAM}
657 \newcommand\wpfig@legend@RM{RM}
658 \newcommand\wpfig@legend@RAM{RAM}
659 \newcommand\wpfig@legend@RAM@expl{\if@RAM R(A)M $\widehat=$ Researcher (Assistant) Months\else\ Efforts in PM
660 \newcommand\wpfig@legend@lead@expl{WP lead efforts \wp@lead@style@explained}
661 \newcommand\wpfig@legend@caption{{\ifwork@areas Work Areas and \fi}Work Packages}
```

```
662 \def\wpfig@style{}
663 \newcommand\wpfigstyle[1]{\def\wpfig@style{#1}}
```

```
664 \newcount\local@count
665 \newcount\@@@RM\if@RAM\newcount\@@@RAM\fi
666 \newcount\all@@@RM\if@RAM\newcount\all@@@RAM\fi
667 \newcommand{\wpfig}[1][]{\setcounter{wpfig@options}{0}\setkeys{wpfig}{#1}
```

the first thing to do is to build the body of the table programmatically by (globally) extending the
\@wp@lines token register inside a bracket group which locally redefines all macros we are using
in the extensions, so that they do not get into the way. We start this group now.

```
668 {\gdef\@wp@lines{}%initialize
669 \let\tabularnewline\relax\let\hline\relax\let\lead@style\relax% so they
670 \let\wa@style\relax\let\wp@style\relax \let\@sw\relax\let\textbf\relax% do not
671 \let\G@refundefinedtrue=\relax\let\@latex@warning=\relax\let\hyperlink=\relax% bother
672 \let\pn\relax\let\xspace\relax% us
```

The code that follows now, could be more elegant, if we had a better way of organizing the data,
but this works for now, we have four cases: with/without work areas and with/without sites. All
do something very similar.

```
673 \ifwork@areas
674 \edef\@@was{\pdataref@safe{all}{wa}{ids}}%
675 \@for\@@wa:=\@@was\do{% iterate over the work areas
676 \xdef\@@wa@line{\wa@style{\pdataRef{wa}\@@wa{label}}%
677 &\wa@style{\@ifundefined{wa@\@@wa @short}{\pdataref{wa}\@@wa{title}}{\pdataref{wa}\@@wa{short}}}%
678 \ifx\wpfig@type\@true&\wa@style{\pdataref{wa}\@@wa{type}}\fi%
679 \ifx\wpfig@pages\@true&\wa@style{\pdataref{wa}\@@wa{page}}\fi%
680 \ifx\wpfig@start\@true&\wa@style{\pdataref{wa}\@@wa{start}}\fi%
681 \ifx\wpfig@length\@true&\wa@style{\pdataref{wa}\@@wa{len}}\fi%
682 \ifx\wpfig@end\@true&\wa@style{\pdataref{wa}\@@wa{end}}\fi}
683 \if@sites
684 \@for\@site:=\prop@gen@sites\do{%
685 \edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}%
686 \local@count 0%
687 \@for\@@wp:=\@@wps\do{\advance\local@count by \pdataref@num\@@wp\@site{RM}}%
688 \pdata@def\@@wa\@site{RM}{\the\local@count}%
689 \xdef\@@wa@line{\@@wa@line&\wa@style{\the\local@count}}%
690 \if@RAM
691 \local@count 0%
692 \@for\@@wp:=\@@wps\do{\advance\local@count by \pdataref@num\@@wp\@site{RAM}}
693 \pdata@def\@@wa\@site{RAM}{\the\local@count}%
```

---

[7] EdNote: document above

[8] EdNote: The computation can be distributed much more efficiently (by intermingling the counter advances with
the row creation), but this works now

```latex
694 \xdef\@@wa@line{\@@wa@line&\wa@style{\the\local@count}}%
695 \fi}
696 \local@count0\relax%
697 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num\@@wa\@site{RM}}%
698 \xdef\@@wa@line{\@@wa@line &\wa@style{\textbf{\the\local@count}}}
699 \if@RAM
700 \local@count0\relax%
701 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num\@@wa\@site{RAM}}%
702 \xdef\@@wa@line{\@@wa@line &\wa@style{\textbf{\the\local@count}}}
703 \fi
704 \else% if@sites
705 \edef\@@wps{\pdataref@safe{all}{wp}{ids}}%
706 \xdef\@@wa@line{\@@wa@line&\wa@style{\pdataref{wa}\@@wa{RM}}
707 \if@RAM&\wa@style{\pdataref{wa}\@@wa{RAM}}\fi}%
708 \fi% if@sites
709 \xdef\@wp@lines{\@wp@lines\@@wa@line\tabularnewline\hline}% add the line for the workarea
710 \edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}%
711 \@for\@@wp:=\@@wps\do{% iterate over its work packages
712 \xdef\@@wp@line{\pdataRef{wp}\@@wp{label}%
713 &\@ifundefined{wp@\@@wp @short}{\pdataref{wp}\@@wp{title}}{\pdataref{wp}\@@wp{short}}%
714 \ifx\wpfig@type\@true&\pdataref{wp}\@@wp{type}\fi%
715 \ifx\wpfig@pages\@true&\pdataref{wp}\@@wp{page}\fi%
716 \ifx\wpfig@start\@true&\pdataref{wp}\@@wp{start}\fi%
717 \ifx\wpfig@length\@true&\pdataref{wp}\@@wp{len}\fi%
718 \ifx\wpfig@end\@true&\pdataref{wp}\@@wp{end}\fi}
719 \if@sites
720 \@for\@site:=\prop@gen@sites\do{%
721 \edef\@@lead{\pdataref@safe{wp}\@@wp{lead}}
722 \edef\@@RM{\ifx\@@lead\@site\lead@style{\pdataref@safe\@@wp\@site{RM}}\else\wp@style{\pdataref@safe\@@wp\@sit
723 \xdef\@@wp@line{\@@wp@line&\@@RM}
724 \if@RAM
725 \edef\@@RAM{\ifx\@@lead\@site\lead@style{\pdataref@safe\@@wp\@site{RAM}}\else\wp@style{\pdataref@safe\@@wp\@s
726 \xdef\@@wp@line{\@@wp@line&\@@RAM}
727 \fi}
728 \local@count0\relax%
729 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num\@@wp\@site{RM}}%
730 \xdef\@@wp@line{\@@wp@line &\textbf{\the\local@count}}
731 \if@RAM
732 \global\local@count0\relax%
733 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num\@@wp\@site{RAM}}%
734 \xdef\@@wp@line{\@@wp@line &\textbf{\the\local@count}}
735 \fi% if@sites
736 \else% if@sites
737 \xdef\@@wp@line{\@@wp@line&\wp@style{\pdataref@safe{wp}\@@wp{RM}}}
738 \if@RAM\xdef\@@wp@line{\@@wp@line&\wp@style{\pdataref@safe{wp}\@@wp{RAM}}}\fi
739 \fi% if@sites
740 \xdef\@wp@lines{\@wp@lines\@@wp@line\tabularnewline\hline}}}
```

 Now the case where we do not have work areas.

```latex
741 \else% ifwork@areas
742 \edef\@@wps{\pdataref@safe{all}{wp}{ids}}%
743 \@for\@@wp:=\@@wps\do{% iterate over its work packages
744 \xdef\@@wp@line{\pdataRef{wp}\@@wp{label}%
745 &\@ifundefined{wp@\@@wp @short}{\pdataref{wp}\@@wp{title}}{\pdataref{wp}\@@wp{short}}
746 \ifx\wpfig@type\@true&\pdataref{wp}\@@wp{type}\fi%
747 \ifx\wpfig@pages\@true&\pdataref{wp}\@@wp{page}\fi%
748 \ifx\wpfig@start\@true&\pdataref{wp}\@@wp{start}\fi%
749 \ifx\wpfig@length\@true&\pdataref{wp}\@@wp{len}\fi%
750 \ifx\wpfig@end\@true&\pdataref{wp}\@@wp{end}\fi}
```

```
751 \if@sites
752 \@for\@site:=\prop@gen@sites\do{%
753 \edef\@@lead{\pdataref@safe{wp}\@@wp{lead}}
754 \edef\@@RM{\ifx\@@lead\@site\lead@style{\pdataref@safe\@@wp\@site{RM}}\else\wp@style{\pdataref@safe\@@wp\@sit
755 \xdef\@@wp@line{\@@wp@line&\@@RM}
756 \if@RAM
757 \edef\@@RAM{\ifx\@@lead\@site\lead@style{\pdataref@safe\@@wp\@site{RAM}}\else\wp@style{\pdataref@safe\@@wp\@s
758 \xdef\@@wp@line{\@@wp@line&\wp@style\@@RAM}
759 \fi}
760 \global\local@count0\relax%
761 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num\@@wp\@site{RM}}%
762 \xdef\@@wp@line{\@@wp@line &\textbf{\the\local@count}}
763 \if@RAM
764 \global\local@count0\relax%
765 \@for\@site:=\prop@gen@sites\do{\global\advance\local@count by \pdataref@num{#1}\@site{RAM}}%
766 \xdef\@@wp@line{\@@wp@line &\textbf{\the\local@count}}
767 \fi
768 \else% if@sites
769 \xdef\@@wp@line{\@@wp@line&\wp@style{\pdataref@safe{wp}\@@wp{RM}}}
770 \if@RAM\xdef\@@wp@line{\@@wp@line&\wp@style{\pdataref@safe{wp}\@@wp{RAM}}}\fi}
771 \fi% if@sites
772 \xdef\@wp@lines{\@wp@lines\@@wp@line\tabularnewline\hline}}
773 \fi%ifwork@areas
```

Now we compute the totals lines in the `\@totals` macros; again there are four cases to consider

```
774 \gdef\@totals{}
775 \ifwork@areas
776 \if@sites
777 \@for\@site:=\prop@gen@sites\do{% iterate over the sites
778 \@@@RM=0\if@RAM\@@@RAM=0\fi
779 \edef\@@was{\pdataref@safe{all}{wa}{ids}}%
780 \@for\@@wa:=\@@was\do{% iterate over the work areas
781 \edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}%
782 \@for\@@wp:=\@@wps\do{% iterate over the work packages
783 \advance\@@@RM by \pdataref@num\@@wp\@site{RM}%
784 \if@RAM\advance\@@@RAM by \pdataref@num\@@wp\@site{RAM}\fi}}
785 \pdata@def{all}\@site{RM}{\the\@@@RM}\if@RAM\pdata@def{all}\@site{RAM}{\the\@@@RAM}\fi
786 \advance\all@@@RM by \the\@@@RM\if@RAM\advance\all@@@RAM by \the\@@@RAM\fi
787 \xdef\@totals{\@totals & \textbf{\the\@@@RM}\if@RAM& \textbf{\the\@@@RAM}\fi}}
788 \xdef\@totals{\@totals & \textbf{\the\all@@@RM}\if@RAM&\textbf{\the\all@@@RAM}\fi}
789 \pdata@def{all}{total}{RM}{\the\all@@@RM}\if@RAM\pdata@def{all}{total}{RAM}{\the\all@@@RAM}\fi
790 \else% if@sites
791 \@@@RM=0\if@RAM\@@@RAM=0\fi
792 \edef\@@was{\pdataref@safe{all}{wa}{ids}}%
793 \@for\@@wa:=\@@was\do{\edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}%
794 \@for\@@wp:=\@@wps\do{% iterate over the work packages
795 \advance\@@@RM by \pdataref@num{wp}\@@wp{RM}%
796 \if@RAM\advance\@@@RAM by \pdataref@num{wp}\@@wp{RAM}\fi}}
797 \pdata@def{all}{total}{RM}{\the\@@@RM}\if@RAM\pdata@def{all}{total}{RAM}{\the\@@@RAM}\fi
798 \xdef\@totals{&\the\@@@RM\if@RAM &\the\@@@RAM\fi}
799 \fi% if@sites
800 \else%i.e. no work@areas
801 \if@sites
802 \@for\@site:=\prop@gen@sites\do{%iterate over the sites
803 \@@@RM=0\if@RAM\@@@RAM=0\fi%
804 \edef\@@wps{\pdataref@safe{all}{wp}{ids}}%
805 \@for\@@wp:=\@@wps\do{% iterate over the work packages
806 \advance\@@@RM by \pdataref@num\@@wp\@site{RM}%
807 \if@RAM\advance\@@@RAM by \pdataref@num\@@wp\@site{RAM}\fi}
```

```
808 \pdata@def{all}\@site{RM}{\the\@@@RM}\if@RAM\pdata@def{all}\@site{RAM}{\the\@@@RAM}\fi
809 \xdef\@totals{\@totals & \textbf{\the\@@@RM}\if@RAM& \textbf{\the\@@@RAM}\fi}
810 \advance\all@@@RM by \the\@@@RM\if@RAM\advance\all@@@RAM by \the\@@@RAM\fi}
811 \xdef\@totals{\@totals &\textbf{\the\all@@@RM}\if@RAM&\textbf{\the\all@@@RAM}\fi}
812 \pdata@def{all}{total}{RM}{\the\all@@@RM}\if@RAM\pdata@def{all}{total}{RAM}{\the\all@@@RAM}\fi
813 \else% if@sites
814 \@@@RM=0\if@RAM\@@@RAM=0\fi
815 \edef\@@wps{\pdataref@safe{all}{wp}{ids}}%
816 \@for\@@wp:=\@@wps\do{% iterate over the work packages
817 \advance\@@@RM by \pdataref@num{wp}\@@wp{RM}%
818 \if@RAM\advance\@@@RAM by \pdataref@num{wp}\@@wp{RAM}\fi}
819 \pdata@def{all}{total}{RM}{\the\@@@RM}\if@RAM\pdata@def{all}{total}{RAM}{\the\@@@RAM}\fi
820 \xdef\@totals{&\the\@@@RM\if@RAM &\the\@@@RAM\fi}
821 \fi% if@sites
822 \fi
```

And we finally have a line for the intended totals which we use in draft mode.

```
823 \gdef\intended@totals{}\gdef\requested@totals{}
824 \if@sites
825 \@for\@site:=\prop@gen@sites\do{
826 \xdef\intended@totals{\intended@totals&\textbf{\pdataref@safe{site}\@site{intendedRM}}}
827 \xdef\requested@totals{\requested@totals&\pdataref@safe{site}\@site{reqPM}}
828 \if@RAM\xdef\intended@totals{\intended@totals&\textbf{\pdataref@safe{site}\@site{intendedRAM}}}\fi}
829 \if@RAM\xdef\intended@totals{\intended@totals&&}\else%
830 \xdef\intended@totals{\intended@totals&}%
831 \xdef\requested@totals{\requested@totals&}%
832 \fi
833 \else% if@sites
834 \xdef\intended@totals{\intended@totals&\textbf{\pdataref@safe{all}{intended}{RM}}}
835 \if@RAM\xdef\intended@totals{\intended@totals&\textbf{\pdataref@safe{all}{intended}{RAM}}}\fi
836 \fi}% if@sites
```

finally, we make all of this into a figure, computing the colspan of the the legend cells for the totals via `\local@count` from the optional columns.

```
837 \local@count\thewpfig@options\advance\local@count by 2
838 \begin{wp@figure}
839 \@wp@lines\hline%
840 \multicolumn{\the\local@count}{|c|}{\prop@legend@totals}\@totals\\\hline%
841 \ifsubmit\else%
842 \multicolumn{\the\local@count}{|c|}{\prop@legend@intendedtotals}\intended@totals\\\hline
843 \multicolumn{\the\local@count}{|c|}{\prop@legend@requestedtotals}\requested@totals\\\hline
844 \fi
845 \end{wp@figure}}
```

and now multilinguality support

```
846 \newcommand\prop@legend@totals{\textbf{totals}}
847 \newcommand\prop@legend@intendedtotals{\textbf{intended totals}}
848 \newcommand\prop@legend@requestedtotals{\textbf{requested totals}}
```

4.11 Gantt Charts

Gantt Charts are done with help of the the `tikz` package. The `gantt` environments pick up on the declared duration of the proposal in months stored in the `\prop@gen@months` macro.

We define the keys for Gantt tables

```
849 \newif\ifgantt@draft\gantt@draftfalse
850 \define@key{gantt}{xscale}{\def\gantt@xscale{#1}}
851 \define@key{gantt}{yscale}{\def\gantt@yscale{#1}}
852 \define@key{gantt}{step}{\def\gantt@step{#1}}
853 \define@key{gantt}{size}{\def\gantt@size{#1}}
```

854 \define@key{gantt}{draft}[true]{\ifsubmit\else\gantt@drafttrue\fi}

Then we define an auxiliary function that provides defaults for these keys and sets the internal macros.

855 \def\gantt@set#1{\gantt@draftfalse\def\gantt@xscale{1}\def\gantt@yscale{.35}\def\gantt@step{3}
856 \setkeys{gantt}{#1}}

Finally, the Gantt Chart environment itself.

gantt   The gantt[⟨keyvals⟩]{⟨height⟩} environment sets up the grid and legend for a gantt chart. The grid is \prop@gen@months wide and ⟨height⟩ high.

857 \newenvironment{gantt}[2][]
858 {\gantt@set{#1}
859 \def\@test{\prop@gen@months@default}
860 \ifx\@test\prop@gen@months
861 \ClassError{proposal}{Need overall project months to draw gantt
862   chart - expect trouble;\MessageBreak specify
863   \protect\begin{proposal}[...,months=??,...] to fix}\fi
864 \@ifundefined{gantt@size}{}{\csname\gantt@size\endcsname}
865 \newdimen\gantt@ymonths
866 \gantt@ymonths=#2 cm
867 \advance\gantt@ymonths by .5cm
868 \begin{tikzpicture}[xscale=\gantt@xscale,yscale=\gantt@yscale]
869 \draw[xstep=\gantt@step,gray,very thin] (0,0) grid (\prop@gen@months,#2);
870 \foreach \x in {0,\gantt@step,...,\prop@gen@months} \node at (\x,\gantt@ymonths) {\x};}
871 {\end{tikzpicture}}

\@action   In this we have used the macro that does the actual painting. \@action{⟨name⟩}{⟨line⟩}{⟨start⟩}{⟨len⟩}{⟨force⟩} creates a gantt node with name ⟨name⟩ in line ⟨line⟩ starting at month ⟨month⟩ with length ⟨len⟩ that is ⟨force⟩ thick.

872 \newdimen\gantt@ymid\newdimen\gantt@yinc\newdimen\gantt@xend
873 \newcommand{\@action}[5]{%
874 \gantt@ymid=#2 cm\gantt@yinc=\gantt@yscale cm
875 \gantt@xend=#3 cm\advance\gantt@xend by #4 cm
876 \advance\gantt@ymid by \gantt@yinc
877 \fill (#3,#2) rectangle +(#4,#5);
878 \node (#1@left) at (#3,\gantt@ymid) {};
879 \node (#1@right) at (\gantt@xend,\gantt@ymid) {};}

\@dependency

880 \def\@dependency#1#2{\draw[->,line width=2pt,color=red] (#1@right) -- (#2@left);}

tt@compute@effort   A helper function that updates the dimension \gantt@effort according to whether the counter \gantt@month is in the range. It is used in \gantt@chart

881 \newcommand\gantt@compute@effort[3]{% start, len, force
882   \@@e=#1\advance\@@e by #2
883   \ifnum\thegantt@month<#1\else
884   \ifnum\thegantt@month<\@@e
885   \gantt@plus=#3cm\advance\gantt@effort by \gantt@plus\fi\fi}

\ganttchart   This macro iterates over the work areas, their work packages, and finally their work phases to use the internal macro \@action. All of this in the gantt setting.

886 \newcommand{\ganttchart}[1][]{\begin{figure}[ht]\centering
887 \gantt@set{#1}
888 \def\gantt@wps{\pdataref@num{all}{wp}{count}}
889 \begin{gantt}[#1]{\gantt@wps}
890   \newcounter{taskwps}\newcount\@@line
891   \edef\@@was{\pdataref@safe{all}{wa}{ids}}

```
892  \ifwork@areas
893  \@for\@@wa:=\@@was\do{% iterate over work areas
894    \edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}
895    \@for\@@wp:=\@@wps\do{% iterate over work packages
896      \stepcounter{taskwps}
897      \@@line=\gantt@wps\advance\@@line by -\thetaskwps
898      \edef\@@tasks{\pdataref@safe\@@wp{task}{ids}}
899      \node at (-1/\gantt@xscale,\@@line) [above=-2pt] {\pdataRef{wp}\@@wp{label}};
900      \edef\@@wphases{\pdataref@safe{wp}\@@wp{wphases}}
901      \@for\@@ft:=\@@wphases\do{%wp-level work phases
902        \decode@wphase\@@ft
903        \@action\@@wp\@@line\wphase@start\wphase@len\wphase@force}
904      \@for\@@task:=\@@tasks\do{% tasks
905        \edef\@@wphases{\pdataref@safe{task}\@@task{wphases}}
906        \@for\@@ft:=\@@wphases\do{%task-level work phases
907          \decode@wphase\@@ft
908          \@action\@@task\@@line\wphase@start\wphase@len\wphase@force}}}}
909  \else% ifwork@areas false
910  \edef\@@wps{\pdataref@safe{all}{wp}{ids}}
911  \@for\@@wp:=\@@wps\do{% iterate over work packages
912    \stepcounter{taskwps}
913    \@@line=\gantt@wps\advance\@@line by -\thetaskwps
914    \edef\@@tasks{\pdataref@safe\@@wp{task}{ids}}
915    \node at (-1/\gantt@xscale,\@@line) [above=-2pt] {\pdataRef{wp}\@@wp{label}};
916    \edef\@@wphases{\pdataref@safe{wp}\@@wp{wphases}}
917    \@for\@@ft:=\@@wphases\do{%iterate over the wp-level work phases
918      \decode@wphase\@@ft
919      \@action\@@wp\@@line\wphase@start\wphase@len\wphase@force}
920    \@for\@@task:=\@@tasks\do{% task-level work phases
921      \edef\@@wphases{\pdataref@safe{task}\@@task{wphases}}
922      \@for\@@ft:=\@@wphases\do{%iterate over the task-level work phases
923        \decode@wphase\@@ft
924        \@action\@@task\@@line\wphase@start\wphase@len\wphase@force}}}
925  \fi% ifwork@areas end
926  \edef\@@deps{\pdataref@safe{all}{task}{deps}}
927  \@for\@@dep:=\@@deps\do{%
928    \@dependency{\pdataref@safe{taskdep}\@@dep{from}}{\pdataref@safe{taskdep}\@@dep{to}}}
```

The next piece of code generates the effort sum table in draft mode

```
929  \ifgantt@draft
930    \newcounter{gantt@month}
931    \newcount\@@e\newdimen\gantt@effort\newdimen\gantt@plus
932    \@whilenum\thegantt@month<\prop@gen@months\do{% step over months
933      \gantt@effort=0cm
934      \ifwork@areas
935      \edef\@@was{\pdataref@safe{all}{wa}{ids}}
936      \@for\@@wa:=\@@was\do{% iterate over work areas
937        \edef\@@wps{\pdataref@safe\@@wa{wp}{ids}}
938        \@for\@@wp:=\@@wps\do{% iterate over work packages
939          \edef\@@wphases{\pdataref@safe{wp}\@@wp{wphases}}
940          \@for\@@ft:=\@@wphases\do{%iterate over the wp-level work phases
941            \decode@wphase\@@ft
942            \gantt@compute@effort\wphase@start\wphase@len\wphase@force}
943          \edef\@@tasks{\pdataref@safe\@@wp{task}{ids}}
944          \@for\@@task:=\@@tasks\do{% iterate over tasks
945            \edef\@@wphases{\pdataref@safe{task}\@@task{wphases}}
946            \@for\@@ft:=\@@wphases\do{%iterate over the wp-level work phases
947              \decode@wphase\@@ft
948              \gantt@compute@effort\wphase@start\wphase@len\wphase@force}}}}
```

```
949    \fill  (\thegantt@month,-5) rectangle +(1,\gantt@effort);
950    \else% ifwork@areas
951    \edef\@@wps{\pdataref@safe{all}{wp}{ids}}
952    \@for\@@wp:=\@@wps\do{% iterate over work packages
953        \edef\@@wphases{\pdataref@safe{wp}\@@wp{wphases}}
954        \@for\@@ft:=\@@wphases\do{%iterate over the wp-level work phases
955          \decode@wphase\@@ft
956          \gantt@compute@effort\wphase@start\wphase@len\wphase@force}
957        \edef\@@tasks{\pdataref@safe\@@wp{task}{ids}}
958        \@for\@@task:=\@@tasks\do{% iterate over tasks
959        \edef\@@wphases{\pdataref@safe{task}\@@task{wphases}}
960        \@for\@@ft:=\@@wphases\do{%iterate over the wp-level work phases
961          \decode@wphase\@@ft
962          \gantt@compute@effort\wphase@start\wphase@len\wphase@force}}}
963    \fill  (\thegantt@month,-5) rectangle +(1,\gantt@effort);
964    \fi% ifwork@areas
965    \stepcounter{gantt@month}}
966  \fi% ifgantt@draft
967  \end{gantt}
968  \caption{\gantt@caption}\label{fig:gantt}
969 \end{figure}}
```

now the multilingual support

```
970 \newcommand\gantt@caption@main{Overview Work Package Activities}
971 \newcommand\gantt@caption@lower{lower bar shows the overall effort \if@RAM (RAM only)\fi per month}
972 \newcommand\gantt@caption{\gantt@caption@main\ifgantt@draft\xspace (\gantt@caption@lower)\fi}
```

\gantttaskchart This macro is a variant of \ganttchart, but it shows the tasks consecutively, as is useful for EU
EdN:9 projects[9]

```
973 \newcommand{\gantttaskchart}[1][]{\begin{figure}[ht]\centering\gantt@set{#1}
974 \def\gantt@tasks{\pdataref@num{all}{task}{count}}
975 \begin{gantt}[#1]{\gantt@tasks}
976   \newcounter{gantt@tasks}\newcount\@@line
977   \edef\@@wps{\pdataref@safe{all}{wp}{ids}}
978   \@for\@@wp:=\@@wps\do{% iterate over work packages
979     \edef\@@tasks{\pdataref@safe\@@wp{task}{ids}}
980     \@for\@@task:=\@@tasks\do{% iterate over the tasks
981       \stepcounter{gantt@tasks}
982       \@@line=\gantt@tasks\advance\@@line by -\thegantt@tasks
983       \node at (-1/\gantt@xscale,\@@line) [above=-2pt] {\taskreflong\@@wp\@@task};
984       \edef\@@wphases{\pdataref@safe{task}\@@task{wphases}}
985       \@for\@@ft:=\@@wphases\do{%iterate over the task-level work phases
986         \decode@wphase\@@ft
987         \@action\@@task\@@line\wphase@start\wphase@len\wphase@force
988       }}}% end all iterations
989    \end{gantt}
990    \caption{\gantt@caption@main}\label{fig:gantt}
991 \end{figure}}
```

4.12 Coherence

\j*

```
992 \newcommand\jpub{\textcolor{\prop@link@color}{\textbf{\large{$\star$}}}}
993 \newcommand\jpro{\textcolor{\prop@link@color}{\textbf{\large{$\bullet$}}}}
994 \newcommand\jorga{\textcolor{\prop@link@color}{\textbf{\large{$\circ$}}}}
```

---

[9]EDNOTE: this should be incorporated with the gantt chart above, but I am currently to scared to do it so close to the deadline

<dl>
<dt>\add@joint</dt>
<dd>\add@joint{⟨first⟩}{⟨second⟩}{⟨sym⟩} adds ⟨sym⟩ to the the \coherence@⟨first⟩@⟨second⟩ macro for the coherence table.</dd>
</dl>

```
995 \newcommand\add@joint[3]{\@ifundefined{coherence@#1@#2}%
996 {\@namedef{coherence@#1@#2}{#3}}%
997 {\expandafter\g@addto@macro\csname coherence@#1@#2\endcsname{#3}}}
```

<dl>
<dt>\prop@joint</dt>
<dd>This iterates over a comma-separated list of names and makes the necessary entries into the coherence table.</dd>
</dl>

```
998 \newcommand\prop@joint[2]{\@for\@first:=#2\do{%
999 \@for\@second:=#2\do{\ifx\@first\@second\else\add@joint\@first\@second{#1}\fi}}}
```

<dl>
<dt>\joint*</dt>
<dd>Now, some instances that use these.</dd>
</dl>

```
1000 \newcommand\jointproj[1]{\prop@joint\jpro{#1}}
1001 \newcommand\jointpub[1]{\prop@joint\jpro{#1}}
1002 \newcommand\jointorga[1]{\prop@joint\jorga{#1}}
```

\coherencematrix

```
1003 \newcommand{\coherencematrix}{
1004 {\let\tabularnewline\relax\let\hline\relax\let\site\relax% so they do
1005  \let\@sw\relax\let\jpub\relax\let\jpro\relax\let\jorga\relax% not bother us
1006 \gdef\@ct@head{}
1007 \@for\@site:=\prop@gen@sites\do{\xdef\@ct@head{\@ct@head &\site{\@site}}}
1008 \gdef\@ct@lines{\@ct@head\tabularnewline\hline\hline} %initialize with head line
1009 \@for\@site:=\prop@gen@sites\do{\xdef\@ct@line{\site{\@site}}
1010   \@for\@@site:=\prop@gen@sites\do{
1011     \xdef\@ct@line{\@ct@line&\ifx\@site\@@site{X}\fi
1012       \@ifundefined{coherence@\@site @\@@site}{}{\@nameuse{coherence@\@site @\@@site}}}}
1013   \xdef\@ct@lines{\@ct@lines\@ct@line\tabularnewline\hline}}}
1014 \begin{tabular}{|l||*{\the@site}{c|}}\hline
1015 \@ct@lines\hline
1016 joint&\multicolumn{\the@site}{l|}{\jpub $\hat=$ publication, \jpro $\hat=$ project,
1017   \jorga $\hat=$ organization}\\\hline
1018 \end{tabular}}
```

\coherencetable

```
1019 \newcommand\coherencetable{%
1020 \begin{table}[ht]
1021 \begin{center}\small\setlength{\tabcolsep}{.5em}
1022 \renewcommand{\arraystretch}{.9}\coherencematrix
1023 \end{center}
1024 \caption{\coherence@caption}\label{tab:collaboration}
1025 \end{table}}
```

now the multilinguality support

```
1026 \newcommand\coherence@caption{Previous Collaboration between {\pn} members}
1027 ⟨/cls⟩
```

4.13   Relevant Papers & References

We first define a bibLaTeX bibliography heading that does not create headers, we need it somewhere.

```
1028 ⟨*cls | reporting⟩
1029 \defbibheading{empty}{}
```

We define an internal macro that prints a publication list of a given bibTeX entry type and title for convenience. It also adds a `notype=` to the token register `\prop@rl` to deal with the unclassified entries from the list.

```
1030 \newif\if@allpapers\@allpaperstrue
1031 \newcommand\prop@ppl[3][]{\@allpapersfalse\message{ppl processing: #2}%
1032 \printbibliography[heading=subbibliography,type=#2,title=#3#1]%
1033 \@ifundefined{prop@rl}{\xdef\prop@rl{#2}}{\xdef\prop@rl{\prop@rl, #2}}}
```

The following code does not work yet, it would have been nice to be able to just add a key `unclassified` to catch the unclassified ones. I guess we just have to issue a warning instead.

```
1034 \newcommand\prop@prl[1]{\message{unclassified: #1}%
1035 \printbibliography[heading=subbibliography,title=Unclassified,#1]}%
1036 \define@key{paperlist}{unclassified}[true]{\message{unclass: \prop@rl}\prop@prl\prop@rl}
```

with this, we define a couple of keys that generate

```
1037 \define@key{paperlist}{articles}[true]{\prop@ppl{article}{Articles}}
1038 \define@key{paperlist}{chapters}[true]{\prop@ppl{inbook}{Book Chapters}}
1039 \define@key{paperlist}{confpapers}[true]{\prop@ppl[,keyword=conference]{inproceedings}{Conference Papers}}
1040 \define@key{paperlist}{wspapers}[true]{\prop@ppl[,notkeyword=conference]{inproceedings}{Workshop Papers}}
1041 \define@key{paperlist}{theses}[true]{\prop@ppl{thesis}{Theses}}
1042 \define@key{paperlist}{submitted}[true]{\prop@ppl[,keyword=submitted]{unpublished}{Submitted}}
1043 \define@key{paperlist}{books}[true]{\prop@ppl{book}{Monographs}}
1044 \define@key{paperlist}{techreports}[true]{\prop@ppl{techreport}{Technical Reports}}
```

featured We introduce a new bibLaTeX category `featured` for those papers that were already mentioned in \prop@paperlist and the macros defined from it.

```
1045 \DeclareBibliographyCategory{featured}
```

\prop@paperlist We generate a subsection with a `refsection` (this makes a separate bibliography for this section) and activate the keys via \nocite. Then we just print the bibliography with the empty header we created before.

```
1046 \newcommand\prop@paperlist[2][]{%
1047 \begin{refsection}%
1048 \nocite{#2}\addtocategory{featured}{#2}%
1049 \let\biboldfont\bibfont%
1050 \renewcommand{\bibfont}{\footnotesize}%
1051 \renewcommand{\baselinestretch}{.9}
1052 \setkeys{paperlist}{#1}
1053 \@ifundefined{prop@rl}{}{\@latex@warning{some papers are not classified!}}
1054 \if@allpapers\printbibliography[heading=empty]\fi%
1055 \let\bibfont\biboldfont%
1056 \end{refsection}}
```

We only have to define the `warnpubs` and `empty` heading constructors

```
1057 \def\prop@warnpubs@message{Many of the proposers' publications are online at one of the following URIs:}
1058 \def\prop@warnpubs@title{References}
1059 \defbibheading{warnpubs}{\section*{\prop@warnpubs@title}%
1060   \@ifundefined{prop@gen@pubspages}
1061 {\@latex@warning{No publication pages specified;
1062                 use the pubspage key in the proposal environment!}}
1063   {\prop@warnpubs@message%
1064 \@for\@I:=\prop@gen@pubspages\do{\par\noindent\csname\@I\endcsname}}}
1065 \defbibheading{empty}{}
1066 ⟨/cls | reporting⟩
```

4.14 Miscellaneous

\signatures

```
1067 ⟨*pdata⟩
1068 \newcommand{\signatures}[1]{\section{#1}
1069 \qquad\number\day. \number\month. \number\year\\[6ex]
```

```
1070 \strut\qquad Date\hfill\@for\@p:=\prop@gen@PIs\do{%
1071 \wa@ref{person}\@p{personaltitle}~\wa@ref{person}\@p{name}\hfill}}
```

\@dmp   The **\@dmp** macro shows metadata information about the keys in the margin if **\keystrue** is specified. This is a debugging tool.

```
1072 \def\@dmp#1{\ifkeys\marginpar{#1}\fi}
```

\euro

```
1073 \renewcommand\euro{\officialeuro\xspace}
1074 ⟨/pdata⟩
```

References

[Koh14a]   Michael Kohlhase. *Editorial Notes for LaTeX*. Tech. rep. Comprehensive TeX Archive Network (CTAN), 2014.

[Koh14b]   Michael Kohlhase. *Preparing DFG Proposals and Reports in LaTeX with `dfgproposal.cls`*. Tech. rep. Comprehensive TeX Archive Network (CTAN), 2014. URL: http://www.ctan.org/get/macros/latex/contrib/proposal/dfg/dfgproposal.pdf.

[Koh14c]   Michael Kohlhase. *`workaddress.sty`: An Infrastructure for marking up Dublin Core Metadata in LaTeX documents*. Tech. rep. Comprehensive TeX Archive Network (CTAN), 2014. URL: http://www.ctan.org/tex-archive/macros/latex/contrib/stex/workaddress/workaddress.pdf.

[Lon]   Brent Longborough. *gitinfo2.sty. A package for accessing metadata from the git dvcs*. URL: http://mirrors.ctan.org/macros/latex/contrib/gitinfo2/gitinfo2.pdf (visited on 10/26/2014).