`workaddress.sty`: An Infrastructure for managing Addresses and Affiliations in LaTeX*

Michael Kohlhase
Jacobs University
http://kwarc.info/kohlhase

John Doe
DFKI
http://dfki.de/jdoe

November 19, 2015

Abstract

The `workaddress` package allows manage addresses and Affiliations in a bibTeX-like manner.[1]

EdN:1

Contents

*Version v0.4 (last revised 2012/09/23)
[1] EdNote: continue

1 Introduction

The `workaddress` package allows manage Addresses and affiliations of persons in
EdN:2 a bibTEX-like manner.[2]

2 The User Interface

2.1 Package Options

`showmeta` The `workaddress` package takes a single option: `showmeta`. If this is set, then the metadata keys are shown (see [Koh15] for details and customization options).

2.2 Database Entries for Persons

The `workaddress` package recognizes that from a metadata perspective, persons are complex entities. In particular, specifying metadata is a tedious and repetitive task that leads to embarrassing errors. Therefore the `workaddress` package takes a hint from bibTeX and allows to specify personal metadata in a database and use it by a database key. The `\WAperson` macro allows to specify personal metadata[3]

```
\WAperson[id=jdoe,affiliation=dfki,department=skss,
          url=http://dfki.de/jdoe]
         {John Doe}
\WAperson[id=miko,affiliation=jacu,department=case,
          url=http://kwarc.info/kohlhase]
         {Michael Kohlhase}
```

Example 1: A small database of Persons

`\WAperson`
EdN:3 with the following keys:

[2] EdNote: continue
[3] EdNote: This should be synchronized with the FOAF specification [BM07]

key	value	comment
id	string	identifier of this person
birthdate	date	birthdate
email		the primary e-mail address
url	URI	primary home page
affiliation	Inst. identifier	the primary professional affiliation
personaltitle	string	the personal title e.g. `King`
academictitle	string	the academic title e.g. `Prof. Dr.`
department	Inst. identifier	the department specified in the work address
workaddress	long string	the work address
privaddress	long string	the private address
worktel	string	work telephone number
privtel	string	private telephone number
workfax	string	work fax number
privfax	string	private fax number
worktelfax	string	if the phone and fax share a prefix, give this as well
privtelfax	string	dito

In Figure 1 we have specified (minimal) metadata for the authors of the `workaddress` package. The metadata can be accessed by specifying the identifiers (given by the `id` key) in the `workaddress` macros defined below, see for instance the `\WAcreators` macro in Figure **??**, which leads to the title block of this note.

Like in bibTeX [Pat], it is a good idea to collect the metadata in a separate file that is input in the document. In practice it may be possible to generate these files from conventional address databases.

2.3 Institutions

Institutions are treated analogously to persons. The `\WAinstitution` macro al-

```
\WAinstitution[id=case,partof=jacu,acronym=CASE,
               url=http://jacobs-university.de/ses/case]
              {Center for Advanced Systems Engineering}
\WAinstitution[id=jacu,url=http://jacobs-university.de]
              {Jacobs University Bremen}
\WAinstitution[id=skss,partof=dfki,url=http://dfki.de/sks,acronym=SKS]
              {Safe and Secure Cognitive Systems}
\WAinstitution[id=dfki,url=http://dfki.de,shortname=DFKI,acronym=DFKI]
              {German Research Center for Artificial Intelligence}
```

Example 2: A small Database of Institutions and their Parts

`\WAinstitution`

EdN:4 lows to specify personal metadata[4] with the following keys:

[4] EdNote: This should be synchronized with the FOAF specification [BM07]

key	value	comment
id	string	identifier of this person
url	URI	primary home page
partof	Inst. identifier	parent institution

2.4 Applications

The data from the address database can be used in various ways. For instance, the `\WAauthorblock` macro creates a block of users and their affiliations. In the context of the database from Figures 1 and 2, `\WAauthorblock{miko,jdoe}` creates

`\WAauthorblock`

Michael Kohlhase John Doe

`\wa@institution@logo` `\wa@institution@logo` creates the logo of an institution from the database, and (if that is not there create a box and a message instead.)

3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the sTeX GitHub repository [sTeX].

1. none reported yet

4 The Implementation

The `workaddress` package generates two files: the LaTeX package (all the code between ⟨*package⟩ and ⟨/package⟩) and the LaTeXML bindings (between ⟨*ltxml⟩ and ⟨/ltxml⟩). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

The general preamble for LaTeXML

```
1 ⟨*ltxml⟩
2 # -*- CPERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Global;
6 use LaTeXML::Package;
```

¡/ltxml¿

4.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. They all come with their own conditionals that are set by the options.

```
7 ⟨*package⟩
```

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{sref}}
\ProcessOptions
⟨/package⟩
```

The first measure is to ensure that the `KeyVal` package is loaded (in the right version). For LaTeXML we also initialize the package inclusions. We need the `pgf` package for the logos.[5]

EdN:5

```
⟨*package⟩
\RequirePackage{sref}
\RequirePackage{pgf}
⟨/package⟩
⟨*ltxml⟩
DeclareOption(undef,sub {PassOptions('sref','sty',ToString(Digest(T_CS('\CurrentOption')))); })
ProcessOptions();
RequirePackage('sref');
⟨/ltxml⟩
```

4.2 Persons

To implement the `\WAperson` macro, we need to implement its keywords.

```
⟨*package⟩
\addmetakey*{wa@person}{id}
\addmetakey*{wa@person}{birthdate}
\addmetakey*{wa@person}{email}
\addmetakey*{wa@person}{url}
\addmetakey*{wa@person}{affiliation}
\addmetakey*{wa@person}{personaltitle}
\addmetakey*{wa@person}{academictitle}
\addmetakey*{wa@person}{department}
\addmetakey*{wa@person}{workaddress}
\addmetakey*{wa@person}{privaddress}
\addmetakey*{wa@person}{worktel}
\addmetakey*{wa@person}{privtel}
\addmetakey*{wa@person}{workfax}
\addmetakey*{wa@person}{privfax}
\addmetakey*{wa@person}{worktelfax}
\addmetakey*{wa@person}{privtelfax}
```

`\wa@def` The next macro is an auxiliary one that puts the value into an appropriate token register.

```
\def\wa@def#1#2#3#4{\expandafter\xdef\csname wa@#1@#2@#3\endcsname{#4}}
⟨/package⟩
```

At the LaTeXML side we have a function `ExportMetadata` that does a similar job, fishing out the metadata keys from the keyval arguments and storing them in a safe place so they can be accessed later.

```
⟨*ltxml⟩
```

[5] EdNote: MK: this may be overkill though

```
sub getKeyValue_noDelim {
my ($keyval,$key)=@_;
my $valuelist = $keyval && ToString($keyval->getValue($key));
$valuelist =~ s/^{(.*)}$/$1/g if $valuelist;
return $valuelist;
}
sub ExportMetadata {
  my $keys = shift;
  my($id, $email,$affill,$address,$url,$name)=$keys
    && map(getKeyValue_noDelim($keys,$_),qw(id email affiliation address url name));
  if ($id) {
   AssignValue('WA_'.$id.'_email',$email,'global') if $email;
   AssignValue('WA_'.$id.'_affiliation',$affill,'global') if $affill;
   AssignValue('WA_'.$id.'_address',$email,'global') if $email;
   AssignValue('WA_'.$id.'_url',$url,'global') if $url;
   AssignValue('WA_'.$id.'_name',$name,'global') if $name;
  } else {print STDERR "Warning: key 'id' undefined in \\WAperson\n"};
  return;}#$
⟨/ltxml⟩
```

`\wa@ref@test` This macro tests whether the information specified is defined, and gives an error message else.

```
⟨*package⟩
\def\wa@ref@test#1#2#3{%
  \@ifundefined{wa@#1@#2@#3}{%
    \PackageError{workaddress}{reference to undefined #3 of #1 #2}%
    {you must define a #1 with #2=#3\MessageBreak%
     via the macro \protect\WA#1, before you can use it!}
  }{}%
}%
```

With this, referencing is simple

`\wa@ref`

```
\def\wa@ref#1#2#3{%
  \wa@ref@test{#1}{#2}{#3}%
  \csname wa@#1@#2@#3\endcsname%
}%
```

With this we can define the `\WAperson` macro, it just clears the keys, sets them again, and stores them in token registers. If course only if a `id` attribute is given, else we raise an error.

`WAperson`

```
\let\wa@persons=\relax
\newcommand\WAperson[2][]{%
  \metasetkeys{wa@person}{#1}%
  \ifx\wa@person@id\@empty%
    \@latex@warning{key 'id' undefined in WAperson}%
```

```
  \else%
    \wa@def{person}\wa@person@id{id}{\wa@person@id}% redundant, but useful for checking
    \wa@def{person}\wa@person@id{name}{#2}
    \wa@def{person}\wa@person@id{email}{\wa@person@email}
    \wa@def{person}\wa@person@id{birthdate}{\wa@person@birthdate}
    \wa@def{person}\wa@person@id{url}{\wa@person@url}
    \wa@def{person}\wa@person@id{affiliation}{\wa@person@affiliation}
    \wa@def{person}\wa@person@id{workaddress}{\wa@person@workaddress}
    \wa@def{person}\wa@person@id{privaddress}{\wa@person@privaddress}
    \wa@def{person}\wa@person@id{personaltitle}{\wa@person@personaltitle}
    \wa@def{person}\wa@person@id{academictitle}{\wa@person@academictitle}
    \wa@def{person}\wa@person@id{department}{\wa@person@department}
    \wa@def{person}\wa@person@id{workaddress}{\wa@person@workaddress}
    \wa@def{person}\wa@person@id{privaddress}{\wa@person@privaddress}
    \wa@def{person}\wa@person@id{worktel}{\wa@person@worktel}
    \wa@def{person}\wa@person@id{privtel}{\wa@person@privtel}
    \wa@def{person}\wa@person@id{workfax}{\wa@person@workfax}
    \wa@def{person}\wa@person@id{privfax}{\wa@person@privfax}
    \wa@def{person}\wa@person@id{worktelfax}{\wa@person@worktelfax}
    \wa@def{person}\wa@person@id{privtelfax}{\wa@person@privtelfax}
    \@ifundefined{wa@persons}{%
      \xdef\wa@persons{\wa@person@id}%
    }{%
      \xdef\wa@persons{\wa@persons,\wa@person@id}%
    }%
  \fi%
}%
\newcommand\DCMperson[2][]{%
  \WAperson[#1]{#2}%
  \PackageWarning{workaddress}{\protect\DCMperson\space is deprecated, use \protect\WAperson\sp
}%
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('wa@person','id','Semiverbatim');
DefKeyVal('wa@person','birthdate','Semiverbatim');
DefKeyVal('wa@person','email','Semiverbatim');
DefKeyVal('wa@person','url','Semiverbatim');
DefKeyVal('wa@person','affiliation','Semiverbatim');
DefKeyVal('wa@person','personaltitle','Semiverbatim');
DefKeyVal('wa@person','academictitle','Semiverbatim');
DefKeyVal('wa@person','department','Semiverbatim');
DefKeyVal('wa@person','workaddress','Semiverbatim');
DefKeyVal('wa@person','privaddress','Semiverbatim');
DefKeyVal('wa@person','worktel','Semiverbatim');
DefKeyVal('wa@person','privtel','Semiverbatim');
DefKeyVal('wa@person','workfax','Semiverbatim');
DefKeyVal('wa@person','privfax','Semiverbatim');
DefKeyVal('wa@person','worktelfax','Semiverbatim');
DefKeyVal('wa@person','privtelfax','Semiverbatim');

```

```
DefConstructor('\WAperson OptionalKeyVals:wa@person {}','',
afterDigest=>sub {
  my ($stomach,$whatsit)=@_;
  my $keys=$whatsit->getArg(1);
  my $name=ToString($whatsit->getArg(2));
  $keys->setValue('name',$name);
  ExportMetadata($keys);
  return;
});#$
⟨/ltxml⟩
```

4.3 Institutions

To implement the `\WAinstitution` macro, we need to implement its keywords first.

```
⟨*package⟩
\addmetakey*{wa@institution}{id}
\addmetakey*{wa@institution}{shortname}
\addmetakey*{wa@institution}{acronym}
\addmetakey*{wa@institution}{url}
\addmetakey*{wa@institution}{partof}
\addmetakey*{wa@institution}{countryshort}
\addmetakey*{wa@institution}{logo}
\addmetakey*{wa@institution}{streetaddress}
\addmetakey*{wa@institution}{townzip}
\addmetakey*{wa@institution}{type}
\addmetakey*{wa@institution}{country}
```

and we proceed as for `\WAperson`,

```
\let\wa@institutions=\relax
```

`WAinstitution`

```
\newcommand\WAinstitution[2][]{%
  \metasetkeys{wa@institution}{#1}%
  \ifx\wa@institution@id\@empty%
    \@latex@warning{key 'id' undefined in WAinstitution}%
  \else%
    \wa@def{institution}\wa@institution@id{id}{\wa@institution@id}% redundant, but useful for c
    \wa@def{institution}\wa@institution@id{name}{#2}
    \wa@def{institution}\wa@institution@id{shortname}{\wa@institution@shortname}
    \wa@def{institution}\wa@institution@id{acronym}{\wa@institution@acronym}
    \wa@def{institution}\wa@institution@id{url}{\wa@institution@url}
    \wa@def{institution}\wa@institution@id{partof}{\wa@institution@partof}
    \wa@def{institution}\wa@institution@id{countryshort}{\wa@institution@countryshort}
    \wa@def{institution}\wa@institution@id{logo}{\wa@institution@logo}
    \wa@def{institution}\wa@institution@id{townzip}{\wa@institution@townzip}
    \wa@def{institution}\wa@institution@id{streetaddress}{\wa@institution@streetaddress}
    \wa@def{institution}\wa@institution@id{country}{\wa@institution@country}
    \wa@def{institution}\wa@institution@id{type}{\wa@institution@type}
```

```
    \@ifundefined{wa@institutions}{%
      \xdef\wa@institutions{\wa@institution@id}%
    }{%
      \xdef\wa@institutions{\wa@institutions,\wa@institution@id}%
    }%
  \fi%
}%
\newcommand\DCMinstitution[2][]{%
  \WAinstitution[#1]{#2}%
  \PackageWarning{workaddress}{\protect\DCMinstitution\space is deprecated, use \protect\WAinst
}%
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('wa@institution','id','Semiverbatim');
DefKeyVal('wa@institution','url','Semiverbatim');
DefKeyVal('wa@institution','partof','Semiverbatim');
DefConstructor('\WAinstitution OptionalKeyVals:wa@institution {}','',
afterDigest=>sub {
  my ($stomach,$whatsit)=@_;
  my $keys=$whatsit->getArg(1);
  my $name=ToString($whatsit->getArg(2));
  $keys->setValue('name',$name);
  ExportMetadata($keys);
  return;
});#$
⟨/ltxml⟩
```

4.4 Applications

`\WAauthorblock` This internal macro builds an author block from a list of `\WAperson` labels in `\wa@creators`.

```
⟨*package⟩
\addmetakey[false]{WAauthorblock}{dept}[true]
\addmetakey[false]{WAauthorblock}{aff}[true]
\addmetakey[false]{WAauthorblock}{url}[true]
\def\@true{true}
\newcounter{authors}
\newcommand\WAauthorblock[2][]{%
  \metasetkeys{WAauthorblock}{#1}
  {\let\tabularnewline\relax
   \@for\@I:=#2\do{\stepcounter{authors}}
   \def\@authors{}%
   \def\@affs{}%
   \def\@depts{}%
   \def\@urls{}%
   \@for\@I:=#2\do{%
     \xdef\@authors{\@authors&\wa@ref{person}\@I{name}}
     \xdef\@@dept{\wa@ref{person}\@I{department}}
     \xdef\@shortname{\csname wa@institution@\@@dept @shortname\endcsname}
```

```
      \xdef\@dept{\ifx\@shortname\@empty\wa@ref{institution}\@@dept{name}\else\@shortname\fi}
      \xdef\@depts{\@depts&\@dept}
      \xdef\@@aff{\wa@ref{person}\@I{affiliation}}
      \xdef\@shortname{\csname wa@institution@\@@aff @shortname\endcsname}
      \xdef\@aff{\ifx\@shortname\@empty\wa@ref{institution}\@@aff{name}\else\@shortname\fi}
      \xdef\@affs{\@affs&\@aff}
      \xdef\@urls{\@urls&\wa@ref{person}\@I{url}}
    }%
    \message{\theauthors authors: \@authors}%
  }%
  \begin{tabular}[t]{l*{\theauthors}{c}}
    \@authors\\
    \ifx\WAauthorblock@dept\@true\@depts\\\fi
    \ifx\WAauthorblock@aff\@true\@affs\\\fi
    \ifx\WAauthorblock@url\@true\@urls\\\fi
  \end{tabular}
}%
⟨/package⟩
%    \begin{macrocode}
% \end{macro}
%
% \begin{macro}{\wapname}
%    \begin{macrocode}
⟨*package⟩
\newcommand\wapname[1]{\wa@ref{person}{#1}{name}}
⟨/package⟩
```

`\waptname`

```
⟨*package⟩
\newcommand\waptname[1]{\wa@ref{person}{#1}{personaltitle} \wa@ref{person}{#1}{name}}
⟨/package⟩
```

`\wa@institution@logo`

```
⟨*package⟩
\newcommand\wa@institution@logo[2][]{%
  \pgfdeclareimage[#1]{logo}{\wa@ref{institution}{#2}{logo}}
  \IfFileExists{\wa@ref{institution}{#2}{logo}}{%
    \pgfuseimage{logo}%
  }{%
    \fbox{#2 logo}\message{still need logo for #2}%
  }%
}%
⟨/package⟩
```

4.5 Finale

Finally, we need to terminate the file with a success mark for perl.

```
⟨ltxml⟩1;
```

References

[BM07] Dan Brickley and Libby Miller. *FOAF Vocabulary Specification 0.91*. Tech. rep. ILRT Bristol, Nov. 2007. URL: http://xmlns.com/foaf/spec/20071002.html.

[Koh15] Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in LaTeX*. Tech. rep. Comprehensive TeX Archive Network (CTAN), 2015. URL: http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf.

[Pat] Oren Patashnik. *bibTeXing*. URL: http://www.ctan.org/get/biblio/bibtex/contrib/doc/btxdoc.pdf (visited on 12/14/2009).

[sTeX] *KWARC/sTeX*. URL: https://svn.kwarc.info/repos/stex (visited on 05/15/2015).