



FP6-IST-507219

PROSYD

Property-Based System Design

Instrument: Specific Targeted Research Project

Thematic Priority: Information Society Technologies

Textual Property-Based Requirements Specification Tool

(Deliverable 1.2/6)

Due date of deliverable: September 1, 2005

Actual Delivery date: September 1, 2005

Start date of project: 01.01.2004

Duration: 3 years

Organisation name of lead contractor for this deliverable: IBM

Revision: 1.1

Project co-funded by the European Commission within the Sixth Framework Programme (2000-2006)		
Dissemination Level		
PU	Public	<input checked="" type="checkbox"/>
PP	Restricted to other programme participants (including the Commission Services)	<input type="checkbox"/>
RE	Restricted to a group specified by the consortium (including the Commission Services)	<input type="checkbox"/>
CO	Confidential, only for members of the consortium (including the Commission Services)	<input type="checkbox"/>

Notices

For information, contact ornia@il.ibm.com.

This document is intended to fulfill the obligations of the PROSYD project concerning deliverable 1.2/6, described in contract number 507219.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

© Copyright PROSYD 2004. All rights reserved.

Table of Revisions

Version	Date	Description and Reason	By	Affected Sections
0.1	24.7.2005	First Version	Eli Bendersky, Avigail Orni	All
0.2	1.8.2005	Updated based on internal IBM review	Avigail Orni	All
1.0	31.8.2005	Approval by project management	C. Eisner	Version number
1.1	20.12.2005	Changes for public version	A. Orni	All

Authors

Eli Bendersky

Avigail Orni

Executive Summary

PSLText is a tool for editing of PSL code[1]. It is part of the Prosyd toolset, and is used in the framework of the Prosyd methodology. PSLText is especially intended for use during the specification stage of the Prosyd methodology, in which the architect augments an English specification document with PSL statements, thus clarifying and formalizing the intention of the English text. These PSL statements are written using the PSLText tool.

PSLText features for supporting this methodology include syntax colouring (displaying code elements in various colours according to functionality), automatic indentation (according to the structure of the code), and one-click insertion of reference comments (for inserting information connecting the PSL code to the English specification text).

PSLText supports two PSL flavours: Verilog flavour and GDL flavour.

PSLText has been implemented as a plug-in for XEmacs, which is an open source text editor. XEmacs supports many designated language modes (for languages such as C, C++, and Lisp). PSLText functions as a PSL language mode within XEmacs. It can be installed as an extension to any existing XEmacs installation.

PSLText is based on a pre-existing XEmacs language mode for the Sugar 2.0 language (in GDL flavour only), which was developed by IBM. The work on PSLText includes two major upgrades to the pre-existing tool:

- Support for Verilog flavour has been added
- The supported language has been upgraded from Sugar 2.0 to PSL

In addition, a user manual has been written for the PSLText tool. A summary of the user manual is included in this document.

Purpose

The purpose of this document is to describe the PSLText tool, which is used within the Prosyd methodology for writing textual property-based requirement specifications in PSL.

Intended Audience

This guide is intended for potential users of the PSLText tool: hardware architects, designers, and verification engineers.

Background

PSLText is part of the Prosyd toolset, and is used in the framework of the Prosyd methodology. Within the flow of the Prosyd methodology, PSLText is intended for use during the specification stage, in which the architect writes a specification document, and augments the English specification with PSL statements. It may also be used during the design and verification phases.

PSLText has been implemented as a language mode plug-in for the XEmacs text editor. XEmacs is an extensible open source text editor, which supports various specialized language modes. PSLText can be installed as an extension to any existing XEmacs installation.

Before the development of PSLText, IBM had developed a similar XEmacs language mode, named "formal-mode", for the Sugar language (the precursor of PSL). The old "formal-mode" supported Sugar 2.0, in GDL flavour only. PSLText is based on the old "formal-mode", with two major upgrades:

- Support for Verilog flavour has been added
- The supported language has been upgraded from Sugar 2.0 to PSL

Contents

Table of Revisions.....	iii
Authors	iii
Executive Summary	iii
Purpose	iii
Intended Audience.....	iv
Background.....	iv
Table of Figures	vi
Glossary	vii
1 Introduction.....	1
2 PSLText User Manual Summary	3
2.1 Description	3
PSLText for Verilog flavour	3
PSLText for GDL flavour	3
2.2 XEmacs background	3
2.3 Starting PSLText	4
Automatic activation.....	4
Manual activation	4
2.4 Writing PSL code	4
2.5 Menu actions	5
2.6 Customization.....	6
Syntax colouring.....	6
Automatic indentation.....	6
Specification comment	6
3 References.....	7

Table of Figures

Figure 1: Syntax coloring and indentation of a PSL file in PSLText.....	5
---	---

Glossary

GDL: GDL (General Description Language) is the modeling language of the IBM verification tools, RuleBase PE and FoCs. Its primary purpose is to describe the environment for formal verification. However, it is also used in conjunction with PSL to aid specification of a design.

Sugar: The language originally used by IBM's RuleBase PE and FoCs tools as a property specification language, which served as the basis for Sugar 2.0 and PSL.

Sugar 2.0: The language proposed by IBM, which served as the basis for the Accellera PSL standard.

PSL: Property Specification Language, the language for specification of designs upon which PROSYD is based.

Verilog HDL: One of two standardized hardware description languages used to specify the structure and behaviour of electronic systems in textual format. Developed in the mid-1980s as a proprietary language and acquired by Cadence Design Systems, it became a de facto industry standard. In the mid-90s Cadence placed it into the public domain and it became a de jure standard promulgated by the Institute of Electric and Electronic Engineers (IEEE). Verilog is also the name of a legacy simulation tool offered by Cadence.

XEmacs: XEmacs is a customizable open source text editor and application development system. It is related to other versions of Emacs, in particular to GNU Emacs. XEmacs is protected under the GNU public license.

1 Introduction

PSLText is an editing tool, which is specifically designed for editing of PSL code [1]. It is part of the Prosyd toolset, and is used in the framework of the Prosyd methodology (described in the Prosyd Integration Plan [2]). Within the flow of the Prosyd methodology, PSLText is especially intended for use during the specification stage, in which the architect writes a specification document, and then augments the English specification with PSL statements (this stage of the methodology is described in Prosyd Deliverable 1.1/1 [3]). PSLText may also be used during the design and verification phases, when additional PSL code may be written manually.

In the specification phase, the architect writes a specification document (in English), describing the required behaviour of the design. In order to clarify and formalize the intention of the English text, the architect writes accompanying PSL statements. These statements are written in a separate PSL file, using the PSLText tool. Each PSL statement is marked with a reference comment, relating it to the relevant section in the English text. In order to facilitate this methodology, PSLText has the following features:

- **Syntax colouring**
Elements of the code are displayed in various colours according to their functionality
- **Automatic indentation**
Lines of the code are automatically indented according to the structure of the code
- **One-click reference comment**
A standard reference comment can be generated with a single click, for inserting information connecting the PSL code to a specification document

These features support quicker and easier writing of PSL code. Using syntax colouring and automatic indentation, the code becomes more readable. In addition, many syntax errors can be detected as soon as they are written.

PSLText supports two PSL flavours: Verilog flavour and GDL flavour. For each flavour, PSLText recognizes the relevant keywords and syntax, and colours the code accordingly.

PSLText has been implemented as a plug-in for the XEmacs text editor. XEmacs is an extensible, customizable open source text editor (see [3][4]). One of the advantages of XEmacs is its specialized support for a variety of programming languages (such as C, C++, and Lisp). For each of these languages, XEmacs has a designated language mode, which typically understands language syntax, and provides automatic indentation features, syntax based highlighting of text, and language-specific menu actions. PSLText functions as a PSL language mode

within XEmacs. It can be easily installed as an extension to any existing XEmacs installation.

Before work on this deliverable began, IBM had developed a similar XEmacs language mode for the Sugar language (which was the precursor of PSL). This language mode was named "formal-mode". When work on PSLText began, the most recent version of "formal-mode" supported Sugar 2.0 (Sugar 2.0 is the version of Sugar that became the basis for PSL [5]). Only the GDL flavour was supported by "formal-mode".

PSLText is based on the old "formal-mode", with two major upgrades:

- Support for Verilog flavour has been added
- The supported language has been upgraded from Sugar 2.0 to PSL

In addition, a user manual has been written for the PSLText tool. The full user manual is available as part of the PSLText package. A shorter summary version is included in this document in Section 2 .

2 PSLText User Manual Summary

2.1 Description

PSLText is a PSL language mode for use with the XEmacs text editor. PSLText is used for writing PSL code in Verilog flavour or GDL flavour.

PSLText automatically indents and colours PSL code according to the syntax of the language.

PSLText for Verilog flavour

For writing PSL code in Verilog flavour, PSLText activates a custom-made XEmacs language mode named `psl-mode-verilog`.

PSLText for GDL flavour

For writing PSL code in GDL flavour, PSLText activates a custom-made XEmacs language mode named `psl-mode-gdl`.

2.2 XEmacs background

XEmacs is an extensible, customizable open source text editor (see [4]). XEmacs supports various language modes (for languages such as C, C++, Lisp, and many others). These language modes provide features such as automatic indentation, syntax-based highlighting of text, and language-specific menu actions.

PSLText is implemented as an XEmacs language mode. In order to use PSLText, an existing installation of XEmacs is required. PSLText is then installed as an extension of the XEmacs installation.

As part of the installation and customization of PSLText, definitions are added to the XEmacs initialization file. Typically, this file is located at `~/.xemacs/init.el`. In this document, we assume that the XEmacs initialization file is named "init.el".

2.3 Starting PSLText

PSLText is activated within the XEmacs editor. It can be activated while a PSL file is being written, or used for viewing an existing file. The activation can be done automatically, based on the filename extension, or manually, for files with arbitrary names.

Automatic activation

PSLText can be activated automatically for PSL files, using filename association. The PSL files are identified by their filename extension. For each of the flavours – Verilog and GDL – the user defines a set of filename extensions to be associated with that flavour. Whenever a file with one of these extensions is opened in XEmacs, PSLText is activated in the mode of the associated flavour. The file appears in the XEmacs window with appropriate syntax colouring and indentation.

The filename association definitions are added to the `init.el` file.

Manual activation

For any file, regardless of the filename extension, the user may choose to activate one of the PSLText modes.

To activate GDL-flavour mode for a file, open the file in an XEmacs buffer, and type `"M-x psl-mode-gdl"` in the file's buffer.

To activate Verilog-flavour mode for a file, open the file in an XEmacs buffer, and type `"M-x psl-mode-verilog"` in the file's buffer.

Following this activation, the text of the file will be coloured according to the syntax of the chosen flavour.

2.4 Writing PSL code

After a PSL file is opened in an XEmacs buffer, and PSLText is activated in that buffer, PSL code can be written using standard XEmacs editing commands.

Syntax colouring is performed on-the-fly, as the code is written. Elements of the code are displayed in various colours according to their functionality: keywords, identifiers, comments, and quoted strings. Among the keywords, different colours are used for PSL directives, PSL temporal operators, and keywords of the underlying HDL (GDL or Verilog).

Lines of the code are indented according to the structure of the code. A line is indented when PSLText detects termination of a statement (when `;"` is typed), or termination of the line.

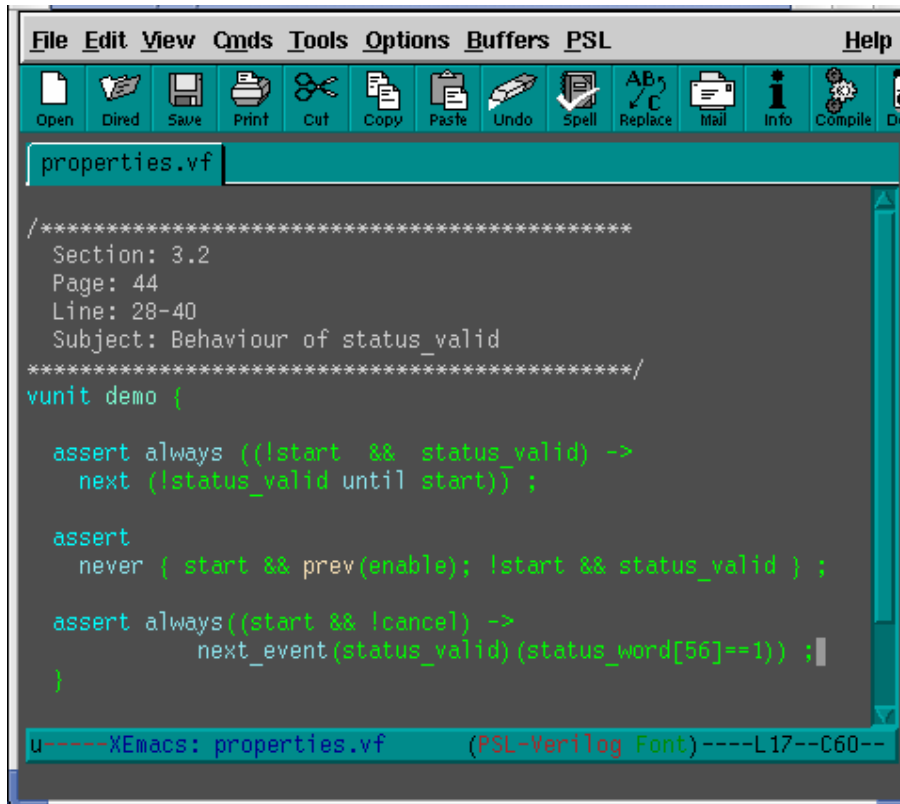


Figure 1: Syntax coloring and indentation of a PSL file in PSLText

2.5 Menu actions

When PSLText is activated in an XEmacs buffer, the PSL menu appears in the menubar. The following actions can then be executed from the PSL menu.

- **Verification unit actions**
 - Mark verification unit
 - Go to beginning of verification unit
 - Go to end of verification unit
- **Automatic specification comment**

Create a comment containing a standard template, for inserting information connecting the PSL code to a specification document.
The text of the template can be customized.
The default comment template is:

```

/*****
Section:
Page:
Line:
Subject:
*****/

```

After inserting the comment template, the user manually adds the reference information.

- **Actions on regions**
 - Comment region
 - Uncomment region
 - Indent region

2.6 Customization

Syntax colouring

The colours that are used by PSLText for highlighting elements of the code can be customized, by setting values in the `init.el` file.

Automatic indentation

Indentation of the code can be customized, by setting values in the `init.el` file.

Specification comment

The text of the comment template, which is generated by the "automatic specification comment" option, can be customized by editing the "psl-mode-auto-comment.el" file, which is part of the PSLText installation.

3 References

- [1] Accellera, Property Specification Language v.1.1, April 2004.
<http://www.eda.org/vfv/docs/PSL-v1.1.pdf>
- [2] "Integration Plan" dated March 2005, as required by Section 9, Item 7 of the October 2004 review report.
- [3] Property-Driven Specification of VLSI design. PROSYD Deliverable 1.1/1.
- [4] <http://www.xemacs.org>
- [5] Sugar 2.0 proposal presented to the Accellera Formal Verification Technical Committee, March 20 2002.
http://www.haifa.il.ibm.com/projects/verification/sugar/Sugar_2.0_Accellera.ps