Edited by H. Ryssel, P. Pichler - September 1995

A Programmable Tool for Interactive Wafer-State Level Data Processing

G. Rieger, S. Halama, and S. Selberherr

Institute for Microelectronics, TU Vienna
Gusshausstrasse 27-29, A-1040 Vienna, Austria

Abstract

This paper presents a tool for initial and intermediate data processing at wafer-state level. Geometric operations and analytical attribute calculations are combined with the graphical user interface and with some TCAD shell functionality to reduce the effort of both manual editing and automated operations within complex simulation flows.

1. Introduction

With the automated simulation of complex process flows some new requirements arose for TCAD frameworks. One of these is the creation of scalable input data for the first simulation step. Others are to perform simple tasks on the actual wafer-state description, like geometry modifications, or to emulate prototype processing steps like mask deposition. As these tasks must be performed in both interactive and batch mode, a highly flexible and versatile implementation is required.

2. The PIF editor

The PED (PIF Editor) has been developed within the VISTA (Viennese Integrated System for TCAD Applications, [4]) framework which is based on the PIF (Profile Interchange Format, [3]). The PED allows creation and modification of device data in batch mode as well as interactively via an X11 Window based graphical user interface. One-, two-, and three-dimensional device geometries can be handled. The basic operations include geometry modeling, attribute handling, and simple visualization. External executables, like, e.g., grid generators, solid modelers, or even simulators can be directly invoked from the PED. This provides an alternative environment for task-level tool integration to the VISTA shell, which is typically used when an immediate interactive access to the output or input of the integrated too is desired.

3. Extension language

A CAD tool like the PIF editor provides functionality for a wide spectrum of tasks. Nevertheless, instead of restricting the user to the functionality provided, it is desirable to adopt the editor to ones needs either by combining basic functions to

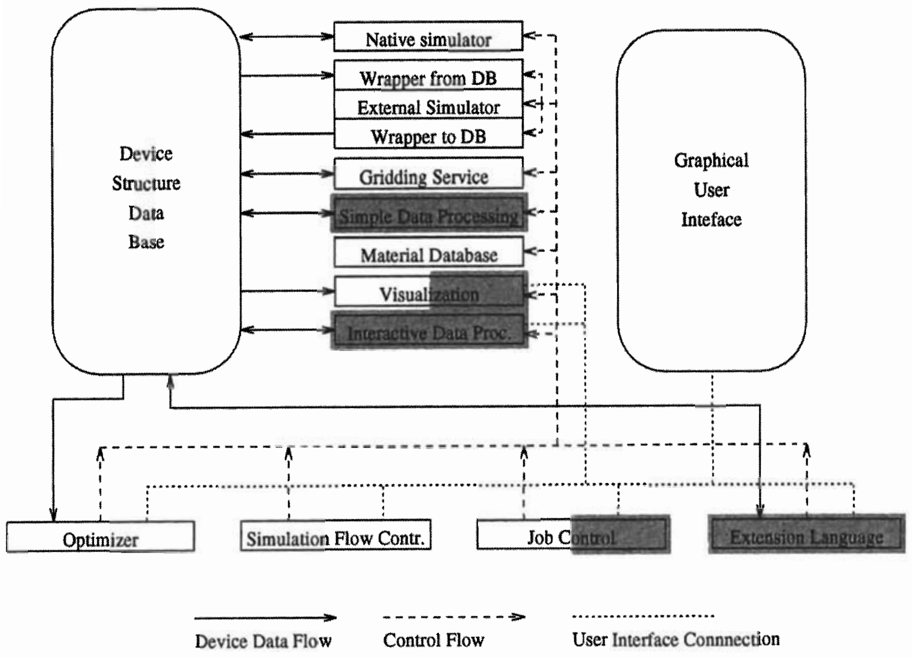


Figure 1: Functional diagram of a TCAD framework and topics met by the PIF editor (shaded)

comfortable editing commands or by generating application–specific facilities. An interpreting extension language is a suitable approach to meet these wishes.

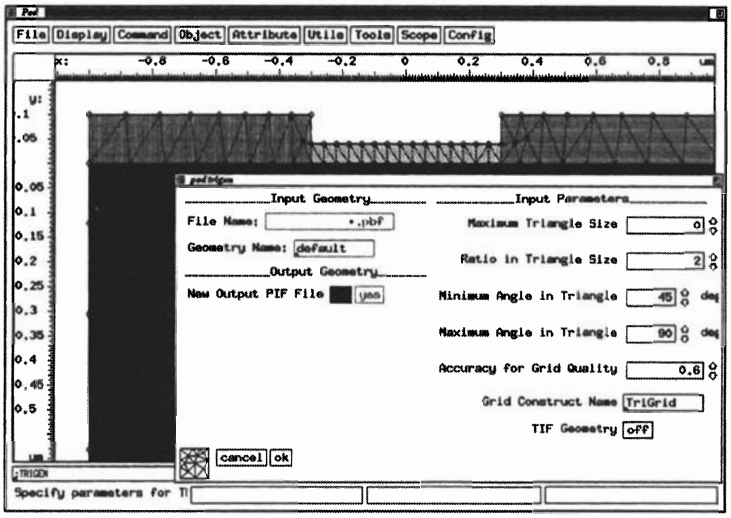
The PED is linked to the VLISP (VISTA LISP) interpreter, a version of the public domain XLISP interpreter, extended with a variety of framework-oriented functions including generic operating system bindings, a comprehensive user interface, and functional access to the PIF database. Utilizing this interpreter makes available a large collection of task level LISP functions already developed for the VISTA shell.

3.1. Configuration and styles

The control and configuration parts of the PED are written in LISP, while low level and time-critical functions are implemented in C, but are bound to the LISP interpreter. Thus the PIF editor can read LISP style- and command-files. Loading the required files makes it possible to build custom versions of the PED that support all geometric elements and attributes required for a specific simulator, whilst allowing the user to focus on only the necessary tasks instead of becoming lost in menus and commands.

3.2. Integration of external executables

The interface for the invocation of the grid generator TRIGEN [2] is shown in Fig. 2. After selection of the program via a menu and filling in the parameters, the TRIGEN executable is started for the device geometry currently displayed. After successful termination of TRIGEN, the result is displayed immediately, allowing for verification and manual improvements of the new grid.


Ped
File
Display
Command
Object
Attribute
Utils
Tools
Scope
Config
x: -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 um
y: .1 .05 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 um
pedtrigen
cancel
ok
;TRIGEN

Input Geometry

File Name: *.pbf

Geometry Name: default

Output Geometry

New Output PIF File yes

Input Parameters

Maximum Triangle Size 0

Ratio in Triangle Size 2

Minimum Angle in Triangle 45 deg

Maximum Angle in Triangle 90 deg

Accuracy for Grid Quality 0.6

Grid Construct Name TriGrid

TIF Geometry off

Specify parameters for TI

Figure 2: PIF editor with the panel for the TRIGEN grid generator, and the resulting grid

3.3. User interface

The usual elements of a graphical user interface - mouse, keyboard, menus and, pop-up windows - are controlled from LISP. All user actions trigger evaluation of LISP expressions which depend on the current style and mode. Operations that require more than one user event are controlled by an infinite state machine [1] that processes an appropriate rule to switch the mode of the PIF editor, to check the input, and to invoke the processing functions.

4. Example Application

A typical example for a PED application is the deposition of a contact and the specification of its electric potential for use by a device simulator.

First a LISP function "`contact`" is written with parameters for contact geometry, position, material, and contact voltage. It generates a rectangular segment and creates appropriate `MaterialType` and `ContactVoltage` attributes. This function can be used, as is, in batch mode. The parameters can be provided by invoking other functions, while "`contact`", which returns a reference to the new segment, might be used in LISP-programs.

"`contact`" can be invoked with the following line:

```
(contact 0.0  -1.0  0.1  0.05 "Al" 0.0)
```

which will generate a rectangular Aluminum segment.

```
(defun contact
  (ybase xmid width height mat voltage
   &aux point1 point2 point3 point4
        segment1 segdesc1 mattype1 contvolt1)

  (setq point1 (point (- xmid (/ width 2)) ybase))
  (setq point2 (point (+ xmid (/ width 2)) ybase))
  (setq point3 (point (+ xmid (/ width 2)) (+ ybase height)))
  (setq point4 (point (- xmid (/ width 2)) (+ ybase height)))

  (setq segment1
        (segment (rectangle)))

  (setq segdesc1  (attribute NIL "SegmentDescription"  segment1))
  (attribute segdesc1 "MaterialType"   segment1 mat)
  (attribute segdesc1 "ContactVoltage" segment1 voltage
             :unit "V")
  segment1)
```

This function can be integrated to the user interface by specification of a rule. This rule lets the state machine inquire the parameters from the user and then invoke the above defined function. The user can specify the values with mouse, keyboard, or by activation of other state machine rules that return appropriate values or objects.

5. Conclusion

The PIF editor is an important tool for the VISTA framework. It allows data processing either program-controlled (for use within extended simulation flows) or interactively via a graphical user interface. It can be tailored to specific requirements with an integrated LISP interpreter.

Acknowledgements

Our work is significantly supported by Sony Corporation, Atsugi, Japan; and Digital Equipment Corporation, Hudson, USA.

References

[1] A.V. Aho, R. Sethi, and J.D. Ullmann, *Compilers*, Addison-Wesley, 1986.

[2] R.E. Bank, "PLTMG: A Software Package for Solving Elliptic Partial Differential Equations", *Frontiers in Applied Mathematics*, vol. 15, SIAM, 1994.

[3] F. Fasching, W. Tuppa, and S. Selberherr, "VISTA-The Data Level", *IEEE Trans. Computer-Aided Design*, vol. 13(1), pp. 72–81, 1994.

[4] S. Halama, F. Fasching, C. Fischer, H. Kosina, E. Leitner, Ch. Pichler, H. Pimingstorfer, H. Puchner, G. Rieger, G. Schrom, T. Simlinger, M. Stiftinger, H. Stippel, E. Strasser, W. Tuppa, K. Wimmer, and S. Selberherr, "The Viennese Integrated System for Technology CAD Applications", In F. Fasching, S. Halama, and S. Selberherr, editors, *Technology CAD Systems*, pp. 197–236, Springer, 1993.