

A Data-Model for a Technology and Simulation Archive

K. Wimmer, M. Noell, W. J. Taylor, and M. Orłowski

Advanced Product Research and Development Laboratory, Motorola Inc.
3501 Ed Bluestein Blvd., Austin, TX 78721, USA

Abstract

In this paper, we describe software aimed at achieving a significant reduction in the time required to develop new semiconductor technologies, by facilitating simulation sharing, and recognition and reuse of technology modules. We have created a technology and simulation archive based on the World-Wide Web (WWW) data-model. The WWW data-model is very flexible in capturing the wide variety of data formats of technology and simulation related information, and it proves to be sufficiently fast for frequent direct data retrieval by simulation tools.

1. Introduction

In a typical design environment for semiconductor technology, engineers of many different skill levels have to navigate through a wide variety of data (simulations, experiments, process flows, etc.) from many different sources.¹

A rigid data-base approach for storage and retrieval of predefined objects is too inflexible to capture our needs:

- facilitate storage, retrieval and reuse of calibrated simulation know-how (input decks, experimental/simulation results, hints and documentation)
- make process flows and recipes directly accessible by simulators
- allow engineers of all levels to acquaint themselves with the technologies and simulation capabilities.

Here we present a complete departure from traditional data bases: An **archive** which is based on the **World-Wide Web (WWW) data-model** [1]. The WWW data-model is a distributed hypermedia system which allows many possible relations (*using hypertext links and index searches*) between any individual objects and others. Each object can be any type of data file that can be stored on a computer.

¹Although there are excellent proposals for a unified data exchange format (e.g. [2]) such a format has not become generally established.

2. Implementation

To use the WWW data-model, one has to set up servers for the `http` [1] and `wais` [3] protocols and to install clients (commonly referred to as browsers) for requesting data from the servers. In this section we focus on selected features in the the WWW data-model which significantly improve its capabilities for a technology and simulation archive.

A browser (most popular browsers are Mosaic and Netscape) needs to know what to do with the data being requested. In order to handle this, each file in the archive is assigned an explicit **content-type** [4]. When a browser requests an object, the server returns the object's content together with its content-type.² Depending on the content-type the browser performs **actions** to view the object. This action, in many cases, is just the presentation in the document window of the browser, but we have found it to be very beneficial to include transformation of the data-formats and spawning of external viewers.

The archive is easily extended to include additional data types by creating a mailcap [5] entry in the browser's `.mailcap` file to associate the content type with the program to view it. For instance, as shown in Figure 1, `tonyplot` is used to view results from Silvaco [6] simulators.

For each object in the archive a **description** is entered. This description is displayed during directory-based browsing to help the user to identify the object. Furthermore we plan to index the descriptions to allow searches.

During check-in of an object into the archive, both the content-type and the description are entered into a local `.htaccess` file. Additionally, new hyperlinks need to be created and indexes need to be updated by corresponding routines on the server.

Uploading of new information and other administrative functions are performed by `http-POSTs` to WWW-agents (CGI-script [7]) from either the browser (using HTML-forms [8]) or separate programs (Figure 2).

```
# Use 'tonyplot' to view ssf.
application/x-ssf;      tonyplot %s
# Convert to ssf and use 'tonyplot'
application/x-pep-sav; pep2ssf -s %s\; tonyplot %s
application/x-pbf;     tifwrap -s %s\; tonyplot %s
application/x-tif;     tif2ssf -s %s\; tonyplot %s
....
```

Figure 1: In the browsers' mailcap file content-types and corresponding viewing actions are specified.



Figure 2: Uploading data to the server is performed by `http-POST` to WWW-agents (CGI-scripts).

Proprietary information is **protected** by a combination of (i) access restriction by domain (Internet address of client), and (ii) a simple user authentication scheme (userid, password) using X.500 directory [7] services. Eventually encryption schemes such as PGP [9] will be included for transfers to off-site locations.

3. Utilization

We have used the WWW data-model to build up a technology and simulation archive which we call the "Simulation and Technology Information Discovery and Access System" (Figure 3). Currently our system provides for browsing (hyperlinked, directory based, and search-able through free text and fielded indexes) of the following kind of semiconductor technology related data:

²Actually, the browser negotiates the acceptable content-types with the server

- Calibrated process and device simulations (input files, results, etc.) for, e.g., PowerPC, fast SRAM (4Mbit, 16Mbit, 64Mbit), non-volatile memory (Flash, EE, 1T, 1.5T cell). These simulations are intended for reuse and to demonstrate current simulation capabilities.
- For simulation automation tools (see below) as well as for training and discovery learning, process flows hyperlinked to miniflows and to technology cross-sections at key steps in the process; individual recipes (miniflows) and masksets.
- Solutions to recurrent simulation issues (e.g. shallow trench and PELOX isolation)
- Manuals, examples and helpful hints
- WAIS index of U.S. patent abstracts; to follow projects complete index of Sematech and SRC abstracts; technical reports, table of contents of selected journals, etc.

We store process flow representation (**PFR**) and recipe (**miniflow**) information in the original format as generated by the **CIM-system** [10], which is **familiar to the engineers**. We have developed a set of supporting **flow-to-<simulator>** tools which translate the PFRs, mask information, and the miniflows into valid simulation input files (Figure 4). The individual tools can be used from the command-line or from VISTA [11] to perform standard device characterizations. The fact that the application engineers now have only to deal with the familiar PFR greatly **enhanced** the **acceptance** of advanced simulators.

Each recipe referenced by the PFR (Figure 5) is stored in a **miniflow** file (Figure 6). The miniflows from the CIM-environment are represented together with process simulator commands to capture knowledge from previous experience. To account for the variety of process simulators (1D vs. 2D, different vendors) the miniflow's command lines are translated to the individual simulator syntax. To adopt to specific process simulator requirements we use **cpp-like** extensions in the miniflows. The masks referenced in the PFR are stored in a maskset file. For each mask and each device cross-section the information about covered and exposed areas is stored in parameterized form (Figure 7).

An issue is the average response time of the server during **frequent accesses** to the archive. The above mentioned translation process by the **flow-to-<simulator>** tools requires to access several hundred objects from the archive server. We experienced, that the response time of the server is short enough to translate typical PFRs into simulator-input files within a few seconds. I.e., the WWW data-model is **sufficiently fast** for frequent direct data retrieval by simulation tools.

4. Conclusion

We have created a technology and simulation archive based on the WWW data-model. The WWW data-model proved its efficiency and flexibility in capturing the wide variety of data formats of technology and simulation related information. Technology knowledge, including simulation, can be **collectively** built up and made available to **individuals**. Due to its open configuration with hyperlinks and index searches, the archive is the perfect environment for **discovery learning** for users of all skill levels. Additionally, the WWW provides a convenient platform for technology transfer while complying with security directives.

References

- [1] T. Berners-Lee, et al.: *Electronic Networking*, 2(1), pp. 52-58, Meckler Publ. 1992.

[2] F. Fasching, et al.: *IEEE Trans. CAD*, 13(1), pp. 72–81, 1994.
 [3] B. Kahle, et al.: *Electronic Networking*, 2(1), pp. 59–68, Meckler Publ. 1992.
 [4] N. Borenstein: Internet Network Working Group, *RFC 1521*, 1993.
 [5] N. Borenstein, et al.: Internet Network Working Group, *RFC 1521*, 1993.
 [6] Silvaco International: *Virtual Wafer Fab User’s Manual*, 1994.
 [7] C. Liu, et al.: *Managing Internet Information Services*, pp. 357–380, O’Reilly & Associates 1994.
 [8] J.E. Tilton: *Composing Good HTML*.
<http://www.willamette.edu/html-composition/strict-html.html>, 1994.
 [9] S. Garfinkel: *PGP: Pretty Good Privacy*. O’Reilly & Associates 1994.
 [10] PROMIS Systems Corp.: *PROMIS User’s Manual*, 1992.
 [11] S. Halama, et al.: *Technology CAD Systems*, pp. 197–236, Springer-Verlag 1993.
 [12] R.W. Dutton, et al.: *Technology CAD: Computer Simulation of IC Processes and Devices*. Kluwer 1993.

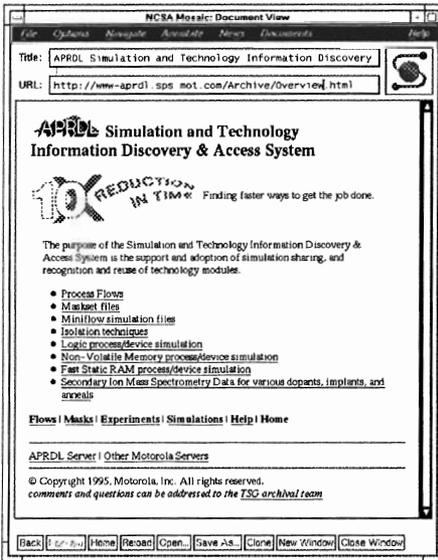


Figure 3: Example home page, the front door to APRDL Simulation and Technology Information Discovery & Access System.

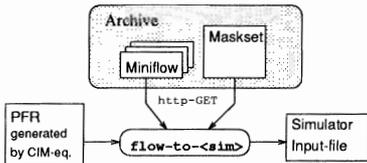


Figure 4: Flow-to-<simulator> tools create input files for various simulators from original CIM-syntax.

```

....
....
14 active PR                               PRxy1A
15 active nit etch 1400A                    ERxy2M
16 n-chan field PR                          P06    PRxy1A
17 n-chan field implant                      B,100keV,1.0e13 IMxy3
18 n-chan field PR strip                    ERxy4
19 field drive/ox                            Dxy5B
....
Miniflow ref#
    
```

Figure 5: Example PFR generated by the CIM-equipment [10]. The process shown is from [12].

```

### Miniflow: Dxy5B - Field Drive & Oxidation
# Push 0:10:00 800 degC N2+2%O2
# Ramp 0:12:00 1000 degC N2+2%O2
# Drive 2:00:00 1000 degC N2+2%O2
# Oxidize 3:10:00 1000 degC O2+H2
# Ramp 0:40:00 800 degC N2
# Pull 0:15:00 800 degC N2
....
diff time=120 temp=1000 dry pressure=0.0196
#ifdef Athena
method viscous oxide.rel=5e-3 grid.oxide=0.05
#endif
diff time=190 temp=1000 wet pressure=0.85
....
    
```

Figure 6: Example of a miniflow file, showing recipe [12], generic simulator commands and simulator specific extensions.

```

Mask#
P01 pmosT # This region left bare after PR
... device-cross-section
P05 nmosT deposit photores thickness=1.0000
P05_nmosT etch photores left x=<length1>
P05_nmosT etch photores right x=<length2>
....
    
```

Figure 7: Maskset: specifies the covered and exposed areas for each mask and device.