システムＬＳＩ設計技術　109－2
(2003. 3. 10)

細粒度リポジトリに基づくVHDLツールプラットフォーム

寺澤　真†　山本晋一郎††　阿草　清滋†

† 名古屋大学大学院工学研究科　〒 464-8603 名古屋市千種区不老町
†† 愛知県立大学情報科学部　〒 480-1198 愛知県愛知郡長久手町熊張茨ヶ廻間
E-mail: †{mtera,agusa}@agusa.nuie.nagoya-u.ac.jp, ††yamamoto@ist.aichi-pu.ac.jp

あらまし　本研究では，VHDL ツールに VHDL 記述の細粒度解析情報を提供するツールプラットフォーム Vapid を提案する．Vapid は，VHDL 記述の解析器，解析結果を格納するデータベース，データベースへのアクセス機能を提供するアクセスルーチンから構成される．従来の VHDL ツールは，コンポネントレベルの粗い粒度の構文要素情報を用いて開発支援を行う．しかし，コンポーネントより細かい粒度での部品化や再利用を行う場合に，細かい粒度で記述を扱うツールが別に必要となる．Vapid は，VHDL ツールに必要な細粒度の情報を提供することで，ツール開発の労力を軽減する．

キーワード　VHDL，ツールプラットフォーム

VHDL tool platform based on a fine grained repository

Makoto TERAZAWA†, Shinichiro YAMAMOTO††, and Kiyoshi AGUSA†

† Depertment of Information Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya-shi, 464-8603 Japan
†† Faculty Information Science and Technology, Aichi Prefectural University
Ibaragabasama Kumabari, Nagakute-cho, Aichi-gun, Aichi-ken, 480-1198 Japan
E-mail: †{mtera,agusa}@agusa.nuie.nagoya-u.ac.jp, ††yamamoto@ist.aichi-pu.ac.jp

Abstract　In this research, we propose Tool-Platform Vapid which provides fine grained information of VHDL description. Vapid has an analyzer for VHDL, a DB stored up analysis and Access Routines which provide functions of access to the DB for VHDL tools. VHDL tools use coarse grained informations of syntactical components such as entities, architectures and packages. However, designers need tools which deal finely with description when reuse in finer grained than component. Our tool platform reduce labor of developping tools since it provides fine grained information required for VHDL tools.

Key words　VHDL, tool platform

1. はじめに

V-LSI 設計者は，プロセス技術の微細化や半導体材料の改良等による技術の進展により，従来より大規模な回路やシステムをより短時間に設計・検証することが求められている．そのため，ハードウェア記述言語 (HDL) が，使われるようになった．特に最近では，SystemC や SpecC 等のソフトウェアのプログラミング言語に近い抽象度の言語も使われるようになってきている．HDL は，低レベルの回路記述から抽象度の高いシステム記述までの幅広い記述言語として使われている．VHDL は，HDL の中で代表的な言語である．

VHDL を用いた開発では，一から VHDL を利用して回路の記述を作成するのではなく，既存のライブラリから必要なコンポーネントを探し出し組み上げて回路を作成する．そのため，開発現場では，コンポーネントの階層構造を理解するために，Hierarchycal Viewer と呼ばれるコンポーネント間の階層関係や結線関係を表示，編集できるツールが用いられる．

一方，function や procedure などのコンポーネントより細かい粒度での部品化や再利用を行ったり，記述の論理的チェックを行う場合がある．既存の Hierarchycal Viewer の多くは，コンポーネント単位で記述を扱うため，このような作業には適さない．細かい粒度で記述を扱うツールが別に必要となる．

また，各種 HDL 用のツールを作成するために，対象言語の字句・構文解析器等の基本的な解析器を個別に作成しなければならない．しかし，その作成には多くの労力がかかる．そのうえ，解析器自体はツールの本質とは関係ないことが多い．様々

なツールに共通して用いられる機能を提供するツールプラットフォームが存在すれば，解析器を各種ツール間で共有することが可能となる．その結果，開発者はツール開発の労力を大幅に削減できる．

本研究では，VHDL ツール作成支援のため，ツールプラットフォーム Vapid を提案する．Vapid は VHDL 記述の解析器，細粒度の構文解析情報を格納するデータベース，データベースに対するアクセス機能を提供するアクセスルーチンから構成される．Vapid は VHDL ツールに細かい粒度の構文情報を提供する．

2. 従来の VHDL ツール

VHDL を用いて V-LSI 開発を行う場合，ライブラリに登録された基本的なコンポーネントをくみ上げてシステムを構築する [1]．そのような開発現場で用いられるツールとして，コンポーネントの階層構造を表示，編集する Hierarchical Viewer と呼ばれるツールがある．多くの Hierarchical Viewer は，コンポーネントの階層構造を理解するために用いられ，システムを構築するサブ回路がどのように関係を持っているか木構造で表示する．

しかし，システムが大規模化するに従い，自分で作成した回路の部品化や再利用による新しいライブラリの構築等のため，コンポーネントより細かい粒度での記述の管理を行う場合がある．既存の Hierarchical Viewer では entity，architecture，package といったコンポーネントレベルの情報を用いる．コンポーネントの階層構造表示などコンポーネント単位での開発支援を行っているため，それらよりも粒度の細かい情報を用いた支援は基本的に不可能である．

そこで以下に既存の VHDL ツールの問題点を挙げる．

- コンポーネントより小さな構成要素の情報を扱っていない．
- コンポーネントをまたぐオブジェクトの定義，出現の相互参照情報が得られない．

1 番目の問題は，コンポーネントより細かい粒度である function や procedure の抽出による部品化や再利用を妨げている．2 番目の問題は，コンポーネント内で完結していないオブジェクトの振る舞いを容易に理解できないため，保守作業の効率が悪くなる．

3. ツールプラットフォーム

各種 VHDL ツールや研究のための試作システムを作成するには，対象 VHDL 記述の字句・構文解析器を持たなければならないため，解析器の作成に多くの労力が必要なことは大きな問題である．解析器の作成は，言語を扱うツールの作成に不可欠ではあるが，VHDL ツール作成の本質的な要素ではないことが多い．様々な VHDL ツールで共通に用いることができる機能を提供するツールプラットフォームが存在すれば，解析器を VHDL ツールから分離することが可能となり，ツール開発者の労力を軽減することができる．

われわれの研究室では，C 言語と Java 言語のための細粒度

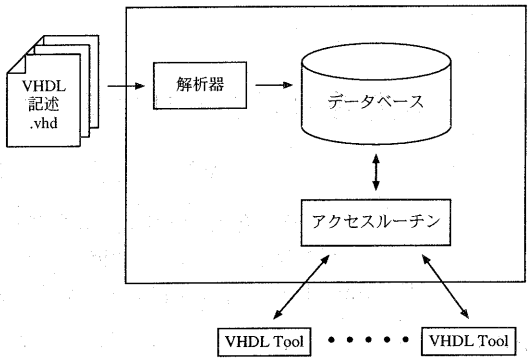

図 1　Vapid の構成
Fig. 1　Structure of Vapid

リポジトリに基づく CASE ツールプラットフォーム Sapid [2] と Japid [3] を提案・実現し，それらを用いてプログラム・スライシングツール，クロスリファレンサ等を作成し，その有効性を示している [4]．

本研究では新たに，VHDL のためのツールプラットフォーム Vapid を作成した．Vapid では Sapid や Japid で培われた技術をハードウェア記述言語 VHDL に応用した．VHDL ツール開発者は，Vapid を利用することで VHDL 記述の細粒度構文解析情報を簡単に得ることができ，新たに解析器を用意することなくツールを開発できる．

4. Vapid

本研究では VHDL 記述を管理することを目的として，VHDL 記述をプログラミング言語のソースプログラムと捉えることで，15 個の実体と 34 個の関連を持つモデル V-model としてモデル化した．さらに，このモデルに従って VHDL 記述を管理するために，VHDL ツールプラットフォーム Vapid を構築した．Vapid は，解析器，データベース，アクセスルーチンで構成される．システムの構成を図 1 に示す．

解析器は入力された VHDL 記述を V-model に基づいて解析する．解析の結果得られた実体・関連情報はデータベースに格納される．アクセスルーチンはデータベースに対するアクセス機能を提供する．VHDL ツールはアクセスルーチンを用いて実現される．

4.1 解 析 器

解析器は，V-model にしたがって入力された VHDL 記述を字句・構文解析し，その結果をデータベースに格納する．

4.2 データベース

データベースには，解析器による字句・構文解析の結果得られた実体・関連情報が格納される．

4.3 V-model

V-model は VHDL 記述の構造を元に VHDL 記述を捉えたモデルであり，VHDL 開発者からの視点を重点に置いている．VHDL の文法をそのまま構造として用いたわけではなく，VHDL 開発者の視点から抽象化を図った．

また，関連の多くは構成関連として捉えることが自然である

表 1　V-model における実体の一覧

Table 1　List of class in V-model

VFile	VHDL のソースファイル
VDesignUnit	library unit と context item からなる design unit
VLibraryUnit	entity,architecture 等の library unit
VContextItem	library 節と use 節からなる context item
VDeclItem	オブジェクトやサブプログラム等の宣言
VIdentifier	オブジェクトやサブプログラム等の識別子
VOptionalDecl	宣言に付随する識別子名以外の情報
VConcurrentStatement	並行文
VSequentialStatement	逐次文
VExpression	式
VLabel	文のラベル
VLiteral	リテラル
VType	type
VBlockConfig	configuration
VOperator	演算子

表 2　V-model における関連の一覧

Table 2　List of relation in V-model

VFileDesign	file を構成する design unit を表す.
VDesignLib	design unit を構成する library unit を表す.
VDesignContext	design unit を構成する context item を表す.
VLibDecl	library unit を構成する宣言を表す.
VArchStat	architecture を構成する並行文を表す.
VEntityStat	entity を構成する並行文を表す.
VRefEntity	entity を実装する architecture を表す.
VDeclType	宣言されているオブジェクトや function の型を表す.
VDeclIdent	宣言されている識別子を表す.
VDeclOpt	DeclItem を構成する OptionalDecl を表す.
VIdentType	識別子の型を表す.
VCompEntity	component 宣言で宣言されている entity を表す.
VInitExpr	初期化式を構成する Expression を表す.
VSubpDecl	サブプログラム内部の宣言を表す.
VSubpBody	サブプログラムを構成する逐次文を表す.
VInterfElement	port 節, generic 節, サブプログラムの引数を表す
VTypeDecl	型宣言を表す.
VBlockConstat	並行文を構成する (サブ) 並行文を表す.
VBlockDecl	並行文内部の宣言を表す.
VProcSeqstat	process 文を構成する逐次文を表す.
VProcDecl	process 文内部の宣言を表す.
VConstatExpr	並行文を構成する式を表す.
VConstatLabel	並行文のラベルを表す.
VBlockSeqstat	逐次文を構成する (サブ) 逐次文を表す.
VSeqstatExpr	逐次文を構成する式を表す.
VSeqstatLabel	逐次文のラベルを表す.
VExprExpr	式を構成する (サブ) 式を表す.
VRefIdent	式を構成する識別子を表す.
VRefLiteral	式を構成するリテラルを表す.
VRefDecl	use 節で参照する宣言を表す.
VConfConf	configuration を構成する block config を表す.
VConfArch	block config が参照する architecture を表す.
VConfLabel	block config が参照するラベルを表す.
VExprOp	式を構成する演算子を表す.

ため，構成関連を的確に表現するモデルとして，Rumbaugh らが提案した OMT モデル [5] のオブジェクト図の記法を採用した．主に用いた記法は，構成関連を表現する記法と汎化を表現する記法である．

図 2 に V-model のオブジェクトモデル図を示す．表 1 に実体の一覧を，表 2 に関連の一覧を示す．

実体と関連の簡単な説明を付録 1. に示す．

4.4　アクセスルーチン

アクセスルーチンは，データベースのデータにアクセスするための C 言語の API である．この API は，VHDL ツール作成者にとって必要かつ基本的な関数を提供する．アクセスルーチンには，データベーススキーマを得るためのクラス・属性名関数，データ属性値取得関数，関連取得関数，実体取得関数などがある．

VHDL ツール作成者はこれらの基本関数を組み合わせてより複雑な機能を作成する．アクセスルーチンを用いることで，

- サブプログラム foo() 内で使われている package 宣言で定義された constant を挙げよ．
- entity bar の宣言部で宣言されたオブジェクトを引数に持つサブプログラムを挙げよ．
- package foobar で定義された signal が使われている式を挙げよ．

などの操作を容易に実現することができる．

表 3　VSpie の生成する情報テーブル

Table 3　Information tables generated by VSpie

	属性
entity	名前，port，generic，local
architecture	名前，entity，port，generic，global，local
package	名前，public，local
signal	名前，type，初期化式
variable	名前，type，初期化式
constant	名前，type，初期化式
type	名前
function	名前，type，引数，global，local
procedure	名前，引数，global，local
configuration	label，entity，architecture

5.　アプリケーション

Vapid の有用性を確かめるため，Vapid を利用したアプリケーションとして VHDL ブラウザ VSpie を試作した．

5.1　VSpie

VHDL ブラウザ VSpie は，VHDL 記述の構文要素間にリンクを張りその関係を明らかにするクロスリファレンス情報をハイパーテキストによって表現するツールである．例えば，指定した entity を component 文で宣言している architecture を見つけだしたり，あるオブジェクトの宣言位置や出現位置の参照を可能にする．既存の VHDL ツールでは，VHDLDOC [6] がある．

VSpie は対象の VHDL 記述を入力すると，WWW ブラウザ上でクロスリファレンスブラウズするための HTML ファイルを生成する．VSpie が出力するファイルには HTML に変換した VHDL 記述の他に，各種情報テーブルがある．この情報テーブルは VHDL 記述から得られる情報をまとめたものである．テーブルから構成要素の定義位置，出現位置をファイル内の行番号及びファイルへのリンクで参照することが可能である．VSpie の生成する情報テーブルを，表 3 に示す．

図 3 は VSpie の表示例である．

6.　評　　価

6.1　Vapid

Vapid は，VHDL 記述の細粒度の構文情報を VHDL ツールに提供する．また VSpie の試作に関しては，VHDL 記述の解析情報を Vapid から得ることにより，VSpie 独自の VHDL 解析器は実装していない．そのため，構文情報から必要な情報 (VSpie ではクロスリファレンス情報) の生成のみ考えればよく，従来のツールと比べると大幅に労力が削減できている．

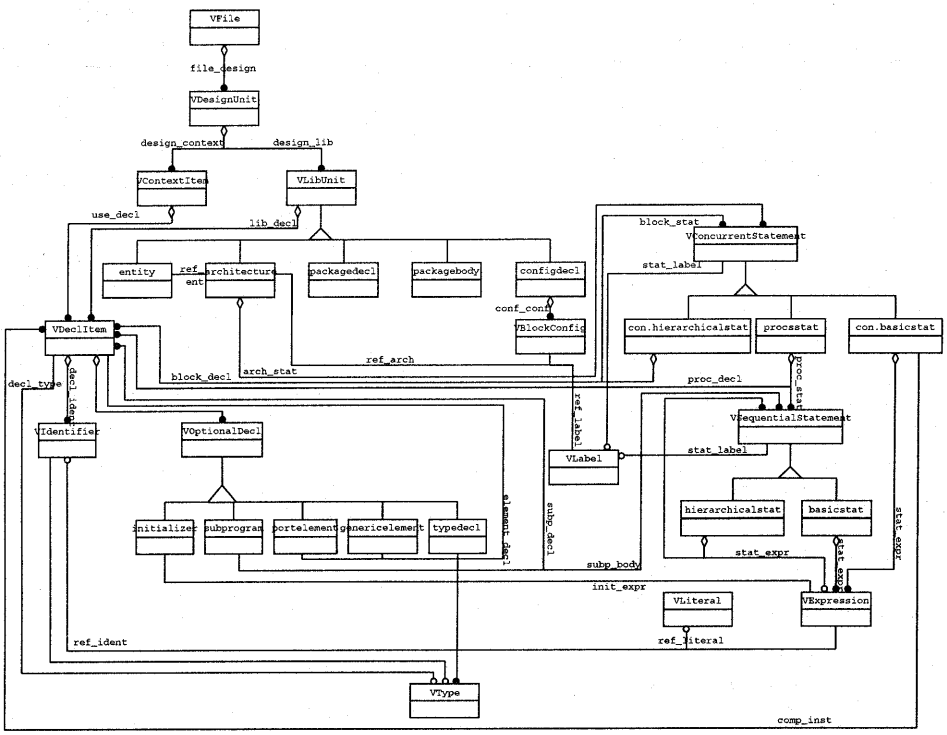

図 2　V-model 図
Fig. 2　Figure of V-mode

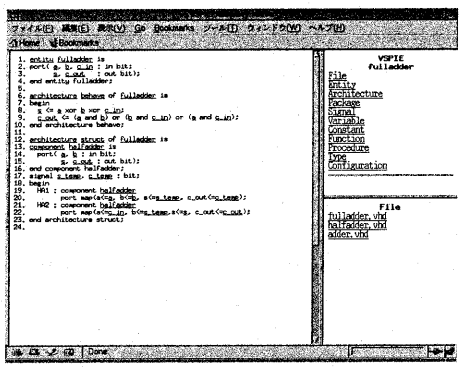

図 3　VSpie
Fig. 3　VSpie

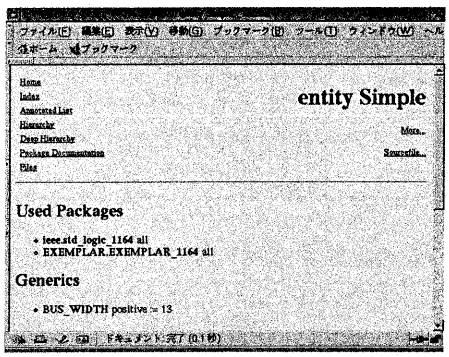

図 4　VHDLDOC
Fig. 4　VHDLDOC

6.2　VSpie

VSpie の評価として既存のツールに VHDLDOC [6] を取り上げ，機能を比較する．

6.2.1　VHDLDOC

VHDLDOC は，VHDL 記述から HTML 文書を自動生成するツールである．生成される HTML 文書は，VHDL 記述の解析により得られる構文情報とコメント中に埋め込まれた特別な記述から生成される．

VHDLDOC は VHDL 記述から複数の HTML 文書群を生成する．それらの HTML 文書群はヘッダとフッタを含む．ヘッダは表 4 に示す項目へのリンクを含む．

entity と architecture について記述するページでは，短い説明文やソース記述へのリンクと長い説明文へのリンクもヘッダに含まれる．

また，フッタには文書の生成に関する情報が含まれる．生成に関する情報とは著者，日付及びそれらの版についての情報である．これらの情報は，所定の場所にタグ付けして記述することで与えることができる．

6.2.2　機 能 比 較

VHDLDOC では，コンポーネントの参照機能があるのに対

表 4 VHDLDOC のヘッダ
Table 4 Headers of VHDLDOC

Index	アルファベット順の entity リスト
Annontated List	短い説明文を含む entity リスト
Hierarchy	階層構造を持った entity リスト
Deep Hierarchy	階層構造を再帰的に持った entity リスト．全ての設計階層が見られる
Package Documentation	package リスト
Files	解析した全てのファイル

表 5 VSpie と VHDLDOC の機能比較
Table 5 Differences of VSpie and VHDLDOC

	VSpie	VHDLDOC
階層構造の閲覧	△	○
ファイル参照	○	○
コンポーネント参照	○	○
サブプログラム参照	○	×
オブジェクト参照	○	×

し，VSpie では，コンポーネント，サブプログラム，オブジェクト，型についての豊富な参照機能を提供しており，従来より細かい粒度で記述を理解することが可能となっている．しかし，VHDLDOC では，コンポーネントの階層構造を木構造で閲覧できるのに対し，VSpie では，ハイパーリンクをたどることで階層構造を理解する必要がある．

VSpie と VHDLDOC の機能の比較を表 5 に示す．

7. おわりに

本研究では VHDL のための VHDL ツールプラットフォーム Vapid を提案した．Vapid は解析器，データベース，アクセスルーチンから構成される．

Vapid の提供する機能を用いて，VHDL ブラウザ VSpie を試作することで，VHDL ツール作成の労力が軽減されることを示した．また，VSpie は細粒度のクロスリファレンス情報を提供することで，VHDL 記述理解支援ツールとして有用であることを示した．

VSpie 以外の EDA ツールを作成し，他のツールでもツール開発労力の低減が図れることを示すこと等の課題が残されている．

文　献

[1] トランジスタ技術，vol.38, no.5, CQ 出版，東京，May. 2001.
[2] 福安 直樹，山本 晋一郎，阿草 清滋，“細粒度リポジトリに基づいた CASE ツール・プラットフォーム Sapid，” 情処学論，vol.39, no.6, pp.1990-1998, Jun. 1998.
[3] Y. Hachisu,S. Yamamoto and K. Agusa, “A CASE tool platform for an object oriented language,” IEICE Trans. Inf. & Syst., vol.E82-D, no.5, pp.997-984, March 1999.
[4] 大橋 洋貴，山本 晋一郎，阿草 清滋，“ハイパーテキストに基づいたソースプログラム・レビュー支援ツール，” 信学技報 SS98-29, pp.15-22, Sep. 1998.
[5] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen, Object-oriented modeling and design, Prentice Hall,New Jersey,1991.
[6] C. Schwick, VHDLDOC, http://schwick.home.cern.ch/schwick/vhdldoc/Welcome.html

付　録

1. V-model 仕様

V-mode では，VHDL 記述をプログラミング言語のソースプログラムと捉えることで，15 個の実体と 34 個の関連を持つモデル V-model としてモデル化した．

以下に，実体と関連に関する簡単な説明を述べる．

1.1 VFile

VHDL の文法規則に則して記述されているファイルを表す．VHDL では，ファイルは複数の design unit からなる．

1.2 VDesignUnit

複数の context item と一つの library unit からなる，design unit を表す．

1.3 VLibraryUnit

entity，architecture，package，configuration 等の library unit を表す．

1.4 VContextItem

library 節や use 節の context item を表す．

1.5 VDeclItem

様々な宣言や本体で宣言される declarative item を表す．

1.6 VIdentifier

declarative item によって宣言される識別子であり，オブジェクトの名前，サブプログラム名，型名等を表す．

1.7 VOptionalDecl

port 節や generic 節に含まれるオブジェクトの宣言，サブプログラム宣言における引数，サブプログラム本体における引数やサブプログラム文部，オブジェクト宣言における初期値式等の様々な宣言に付随する識別子名以外の情報を表す．

1.8 VConcurrentStatement

並行文を表す．

1.9 SequentialStatement

逐次文を表す．

1.10 VExpression

式を表し，複数の式や演算子から構成される．また，その式が function 呼び出し，オブジェクト，リテラルの参照の場合には対応する識別子やリテラルへの参照関係を持つ．

1.11 VLabel

文のラベルを表す．

1.12 VLiteral

リテラルを表す．

1.13 VType

function 定義や様々なオブジェクトの宣言に用いられる型を表す．

1.14 VBlockConfig

component instantiation 文にあらわれる entity と architecture を関連づける configuration を表す．

1.15 VOperator

論理演算子，算術演算子などの演算子を表す．

1.16 関　連

関連はすべて 2 つの実体間の関係を表す．