

УДК 004.8

А.В. Григорьев, канд. техн. наук, доцент,
Д.А. Грищенко, аспирант,
Донецкий национальный технический университет, г. Донецк, Украина
grigorie@pmi.dgtu.donetsk.ua, darya.grischenko@gmail.com

Анализ методов извлечения знаний о методиках проектирования из VHDL-текстов

В статье отмечена актуальность задачи извлечения знаний о методиках проектирования на языке VHDL, рассмотрены существующие методы извлечения знаний и возможные пути извлечения знаний о методиках проектирования в интеллектуальной надстройке над САПР VHDL.

Ключевые слова: VHDL, САПР, интеллектуальная надстройка, методы извлечения знаний, лексический анализ, синтаксический анализ, естественно-языковые текст, формальные языки

Введение

В условиях интенсивного роста сообществ разработчиков радиоэлектронной аппаратуры, их взаимодействия посредством интернет, обмена опытом и решениями типовых задач, возникли огромные массивы VHDL-кода, доступные под свободными лицензиями. Кроме того, каждый разработчик имеет свой багаж готовых решений, апробированных на практике. Поэтому задача анализа и структурирования такого рода массивов кода приобретает особую актуальность в связи с возможностью использования этого багажа знаний в процессе обучения молодых специалистов и ускорения разработки новых проектов [1].

Существующие проектные решения создаются различными авторами, каждый из которых обладает индивидуальными особенностями проектирования вычислительных устройств. Кроме того, одинаковые по функциональности VHDL-модули в разных источниках, могут быть описаны с использованием различных идентификаторов программ. Таким образом, VHDL-программы можно рассматривать не только как код на формальном языке, но и, как текст, содержащий некую субъективную компоненту. В связи с чем, возникает необходимость синтаксического и семантического анализа VHDL-программ, с целью извлечения особенностей методики проектирования устройства конкретным пользователем. Такого рода анализ позволит обеспечить адаптацию САПР к стилю конкретного проектировщика, т.е. позволит учитывать все особенности проектирования устройств разработчиком, а также упростит понимание сгенерированного САПР кода

разработчиком.

Решение задачи анализа больших массивов кода на VHDL возможно при помощи создания интеллектуальной надстройки над САПР VHDL.

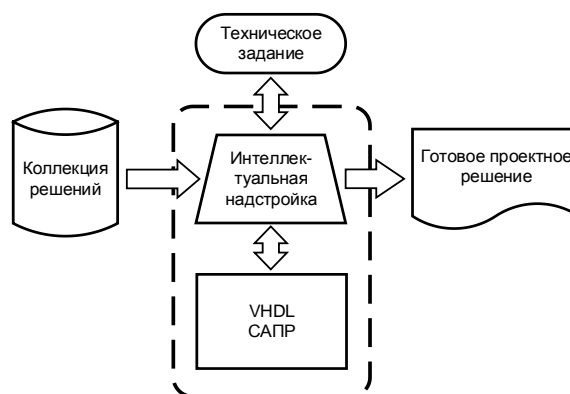


Рисунок 1 - Схема работы интеллектуальной надстройки

Такая надстройка (рис. 1) на входе получает «коллекцию готовых решений» (проектов) пользователя-проектировщика и «Техническое задание» (требования к необходимой разработке), которое формируется в режиме диалога с пользователем. В качестве результата интеллектуальной надстройкой формируется необходимое проектировщику решение на VHDL, максимально соответствующее его стилю проектирования.

Интеллектуальную надстройку будем рассматривать как экспертную систему. В связи, с чем возникает задача создания и наполнения базы знаний, которая будет содержать информацию об используемых в VHDL-решении блоках, их входах/выходах, связях между блоками и их входами/выходами, а также их имена. Такого рода

информацию будем называть прототипом решения. Сопоставление прототипов позволит определить сходства и различия между ними, что в некоторой степени отобразит методику проектирования разработчиком определенного класса устройств.

Таким образом, необходим механизм, позволяющий построить модель методики проектирования, путем извлечения знаний из имеющихся у разработчика готовых решений.

Цель работы - выбор метода извлечения знаний для создания базы знаний интеллектуальной надстройки над САПР VHDL.

Задачи работы - анализ существующих методов извлечения знаний и возможности их применения на этапе создания базы знаний интеллектуальной надстройки над САПР VHDL.

Методы извлечения знаний

Классификация методов извлечения знаний была приведена в работах Гавриловой Т.А. [2,3], где они делятся на коммуникативные и текстологические. Коммуникативные методы подразумевают личное общение с экспертом предметной области и не применимы к решаемой задаче. Текстологические методы анализа информации и извлечения из нее знаний в последнее время получили большее развитие.

Рассмотрим текстологические методы. Основные методы извлечения знаний, их достоинства и недостатки приведены в таблице 1. Наиболее актуальными или развиваемыми в настоящее время являются методы, использующие нейросети, деревья решений и генетические алгоритмы.

В качестве основных задач извлечения знаний выделяют следующие: кластеризация; классификация; прогнозирование; описание данных; задача поиска связанных признаков отдельных документов [4]. Часто, задачи извлечения знаний требуют комбинации различных методов.

Как видно из табл. 1, деревья решений позволяют решать не только задачи извлечения знаний (классификация, прогнозирование), но задачу представления знаний. Наиболее известные методы, основанные на деревьях решений, это:

- CART (Classification and Regression Tree),
- C4.5,
- QUEST (Quick, Unbiased, Efficient Statistical Trees).

Метод CART, разработанный Л. Брейманом и др., представляет собой алгоритм построения бинарного дерева решений и позволяет решать задачи классификации и предсказания [4].

Метод C4.5 – алгоритм построения дерева решений с неограниченным количеством

потомков у узла, разработанный Р. Куинленом, решает задачу классификации.

QUEST – программа, разработанная В. Ло и И. Ши, содержит ряд средств повышения надежности и эффективности индуцируемых деревьев классификации [4].

В [5] рассматриваются такие методы как метод ДРЕВ, AMX (алгоритм, основанный на метрике Хемминга) и ID3. Первые два из них строят бинарные деревья. Метод ID3 обеспечивает высокий уровень классификации и может работать с зашумленными данными, а также имеет ряд модификаций (IDR5, EG2 и др.).

Наиболее важным и проблематичным для методов использующих деревья решений является процесс выбора критерия останова деления на группы, а также процесс отсека незначимых ветвей дерева.

Дивизивные и агломеративные методы извлечения знаний основаны на процессах расщепления или обобщения соответственно. К расщепляющим методам относят различные модификации методов:

- К-средних, где К — количество кластеров на которые необходимо разбить данные;
- С-средних, где входными параметрами являются число кластеров и степень нечеткости.

Среди объединяющих методов выделяют:

- дисперсионные методы иерархической кластеризации;
- методы связи, основанные на объединении объектов в кластер, исходя из вычисленного расстояния между ними;
- центроидные методы, где расстояние между двумя кластерами - это расстояние между их центроидами.

Метод ближайших соседей основан на оценивании сходства объектов. Существуют различные модификации метода, которые в той или иной мере решают проблемы, описанные как недостатки в табл. 1. Наиболее известными модификациями метода являются:

- метод К ближайших соседей, объекты относятся к тому классу, к которому принадлежит большинство его соседей – К ближайших к нему объектов обучающей выборки;

- метод взвешенных ближайших соседей, объект относится к классу, который обладает большим весом среди К ближайших соседей.

К методам, базирующимся на понятии меры близости, также относят метод порогового значения и его модификацию - метод MAXMIN.

Метод корзины покупателя используется для поиска групп характеристик, наблюдаемые одновременно, и дает хорошие результаты, если различные параметры наблюдаются в практически одинаковом числе случаев.

Таблиця 1 – Классификация методов извлечения знаний

Методы	Задача	Достоинства	Недостатки
Деревья решений	Классификация, описание данных, прогнозирование, представление знаний	- оптимизация; - обобщение; - быстрота и эффективность реализации; - наглядность;	- проблема выбора критерия останова процесса деления на группы; - проблема отсека незначимых ветвей; - построение слишком больших деревьев.
Дивизивные и агломеративные	Кластеризация	- используются на начальных этапах; - используются в комплексе с другими методами.	- зависимость результатов от исходного преобразования данных; - зависимость результатов от выбора параметров алгоритма расщепления/объединения; - отсутствие способа проверки правильности разбиения.
Метод корзины покупателя	Ассоциация	- эффективен при исследовании временных рядов; - используется если несколько событий связаны с друг другом.	- резкий рост объема вычислений с увеличением количества параметров; - практически не учитываются редко встречаемые параметры; - ограниченные возможности по учету дополнительных знаний о свойствах параметров.
Нейросети: - обучение с учителем - обучение без учителя	Классификация, Прогнозирование Кластеризация	- адаптируемость; - распределенная ассоциативная память; - обобщение; - отказоустойчивость; - параллельная обработка	- трудность оценки статистической значимости получаемых моделей; - непрозрачность.
Регрессионные методы	Прогнозирование Классификация	- позволяют обнаружить числовые зависимости между данными - позволяют оценить значимость переменных входящих в регрессионную модель	- не все зависимости могут быть обнаружены; - характерны для данных, содержащий большое количество чисел и дискретных значений.
Метод ближайших соседей	Классификация	- простота классификации; - быстрая и эффективная реализация.	- одна размерность у всех параметров; - проблема обработки и хранения больших выборок; - необходимость отсева шума; - проблема выбора метрики.
Генетические алгоритмы	Классификация и кластеризация	- не накладывают ограничения на пространство поиска - позволяют выбрать лучшие (значимые) параметры	- зависимость результата от выбора первоначальной популяции хромосом

Большой класс методов извлечения знаний представлен нейросетями, которые делятся на обучаемые с учителем и без него. Преимуществом сетей, обучаемых без учителя, (сети Кохонена) является их «самостоятельность», отсутствие внешнего классификатора — человека.

Регрессионные методы извлечения знаний чаще всего решают задачу прогнозирования и основаны на статистических методах. Эти методы

позволяют определить зависимости между числовыми данными. К регрессионным методам относят линейную и нелинейную регрессии, логистическую регрессию и т.д. Как правило, регрессионные методы используются на начальных этапах извлечения знаний и применяются к числовым данным.

Генетические алгоритмы способны решать задачи классификации и кластеризации. Чаще

всего, для решения задачи классификации эти алгоритмы используются в комплексе с другими методами, например, используются в целях обучения нейронной сети [6]. В целом, это мощный механизм, который позволяет выделять наиболее значимые параметры или характеристики данных.

Вагин В.Н. [5] в зависимости от объектов обучающей выборки делит методы на структурно-логические и признаковые. К структурно-логическим методам относятся методы, которые в качестве обучающей выборки используют объекты содержащую некоторую внутреннюю логическую структуру (ДСМ-метод, метод ГУНА), соответственно признаковые методы предусматривают работу с объектами, описание которых заданно при помощи признаков.

Наиболее полный обзор методов, решающих задачи классификации, кластеризации и предсказания представлен в [7].

Таким образом, существует огромное количество методов извлечения знаний из текстов, каждый из которых обладает своими достоинствами и недостатками и применяется для определенного класса задач. Кроме того, используются различные подходы к извлечению знаний из естественно-языковых текстов и текстов, которые написанных на формальных языках.

Обработка естественно-языковых текстов

Наиболее актуальным направлением в извлечении знаний из естественно-языковых текстов является извлечение знаний из различных WEB-ресурсов. Причем, если ранее наблюдалась тенденция использования подобного рода технологий в рамках поиска какой-либо информации, то сейчас наблюдается активная работа в рамках определения тональности текстов (нравится/не нравится, хороший/плохой и т.п.). Наиболее популярными инструментами определения тональности текстов являются Twitter Sentiment, I-Teco, Sentiment Analysis with Python NLTK Text Classification и др. Ранее в большинстве случаев в качестве естественно-языковых текстов рассматривались слабоструктурированные тексты, в настоящее время методы извлечения знаний применяются и к хорошо структурированным текстам (решаются задачи реферирования текста и поиска) [8].

Извлечению знаний предшествует этап предварительного анализа текста в процессе, которого осуществляется отбор необходимых текстов, их структуризация и формирование хранилищ исходной информации. Такого рода автоматический анализ текста называют ETL-процессами (extract, transfer, load - извлечение, преобразование, загрузка) [9]. Следующий этап подразумевает

использование методов Data Mining.

В работе [12] авторы выделяют два подхода к решению задачи извлечения информации из текста: символьный и численный или вероятностный подход. Авторы приводят основные методы извлечения данных из ЕЯ текстов, при условии, что факты предметной области представлены фреймами. Это методы извлечения данных на основе:

- признаков (feature-based);
- ядер (kernel-based);
- сопоставления образцов (pattern-based);
- фразовых образцах (phrase-based).

К методам извлечения на основе признаков (feature-based) относят классификаторы Байеса и Скрытые Марковские модели. Суть метода заключается в поиске фрагмента текста, по анализу вероятности признаков, находящихся в окрестности этого фрагмента.

В качестве ядер в kernel-based методах определяется некоторая функция, которая учитывает сложное расположение распознаваемых фрагментов текста, например, в виде деревьев. Таким образом, оценивается вложенность деревьев.

Pattern-based методы основаны на сопоставлении сегментов текста с шаблонам (словам, символам и т.д.).

Методы, основанные на фразовых образцах (phrase-based), являются симбиозом kernel-based и pattern-based методов.

В системах извлечения знаний из ЕЯ-текстов важным этапом является создание словарей терминов. Этот процесс состоит из следующих этапов:

- анализ текстов экспертами (выбор терминов и составление иерархии понятий);
- расширение результатов экспертного анализа путем автоматического извлечения из текстов потенциальных терминов-дескрипторов;
- установление ассоциативных связей между понятиями из словаря.

Чаще всего в качестве словаря с описанием иерархической структуры взаимосвязей между терминами предметной области используются онтологии. В работе [10] такие онтологии определяются как онтологии в «слабом смысле». Такие онтологии могут использоваться для создания поисковых запросов и автоматической классификации данных.

Для выделения терминов предметной области или набора ключевых слов достаточно часто используются подход, основанный на частотном анализе встречаемости слов. В работе [11] для извлечения знаний использовались такие методы как «Метод скольжения открытой рамкой (Slide and trim)» и метод шаблонов. Метод Slide and trim позволяет выделять из текста простые термины и фрагменты предложения, включающего этот

термин, ограниченные специальными словами-пунктуациями. Метод шаблонов использовался для извлечения сложных терминов, состоящих из комбинации простых названий. Этот метод позволяет провести иерархическую классификацию терминов путем установления отношений типа часть и целое. При удачной реализации автоматического процесса создания словарей, их обновление может осуществляться путем обработки новых текстов предметной области.

Создание словарей и использование необходимых для этого методов - это часть работы над ЕЯ-текстами. Системы обработки естественного языка осуществляют различные виды его анализа: синтаксический, семантический, морфологический и т.д.

Для оценки эффективности методов извлечения знаний используют такие критерии как – полнота и точность методов. Чаще всего, точность метода – величина, характеризующая отношение числа правильно отнесенных фактов к заданному классу к общему числу фактов или документов этого класса. Полнота же отражает отношение числа правильно распознанных объектов класса к числу объектов класса в коллекции.

Таким образом, для извлечения знаний из ЕЯ-текстов используются как специфические методы, так и методы Data Mining. Зачастую использование того или иного метода зависит от метода представления знаний и специфики предметной области.

Анализ формально-языковых текстов

Специфика работы с формальными текстами зависит от избранного типа грамматики – контекстно-свободной (КСГ), атрибутивной, предикативной или трансформационной. Можно отметить, что методы работы с КСГ являются основой для работы с грамматиками любых типов. В рамках работы с КСГ существует большое количество инструментов - лексических (ЛА) и синтаксических (СА) анализаторов (сканеров и парсеров). Наиболее популярными инструментами создания лексических и синтаксических анализаторов можно назвать: Lex, Yacc, Flex, Bison.

Lex – генератор лексических анализаторов. Для работы с Lex необходимо создать Lex-спецификацию, состоящую из списка правил. Правила определяют порядок символов, которые надо искать во входном тексте, и что нужно делать, если подходящие выражения найдены. Результатом работы генератора является код на языке Си, после компиляции, которого генерируется программа, осуществляющая лексический анализ. Далее сгенерированная программа, получая на вход исходный файл, формирует список лексем.

Как правило, Lex используется совместно с Yacc (генератор синтаксических анализаторов). На вход генератора синтаксических анализаторов подается файл спецификаций, содержащий описание грамматики необходимого языка. Результатом работы программы являются три файла, один из которых содержит сгенерированную программу анализатора.

Flex (Fast Lexical Analyzer) - также как и Lex является генератором лексических анализаторов и является его аналогом.

Bison – генератор синтаксических анализаторов по заданному описанию грамматики, генерирует программы на языке C, C++, Java. Bison совместим с Yacc и преобразует описание контекстно-свободной грамматики в программу на языке C, для синтаксического разбора этой грамматики.

Так как лексический анализатор позволяет выделять из текста лексемы, то обычно он применяется перед синтаксическим анализом, определяющим соответствие текста определенной контекстно-свободной грамматике. Однако, возможен и вариант одновременного использования лексического и синтаксического анализатора. Результатом работы синтаксического анализатора является дерево, отражающее структуру языка. В зависимости от того, каким образом строятся узлы дерева (от корня к листьям или наоборот), выделяют два класса синтаксических анализаторов: нисходящие и восходящие. Нисходящие анализаторы связаны с LL-грамматиками, которые могут быть проанализированы без возвратов, легко обобщаются, входная цепочка строится слева направо, строится левый вывод цепочки. Восходящие анализаторы связаны с LR-грамматиками (цепочка просматривается слева направо, строится правый вывод цепочки). Восходящие анализаторы позволяют определить большее количество используемых языков программирования.

В рамках решаемой задачи синтаксические и лексические анализаторы можно использовать в качестве предварительного этапа извлечения знаний из VHDL-текстов, как это было сделано в работе [13], где авторы работали над созданием парсера языка VHDL.

Для извлечения знаний из формальных текстов автором [14] ранее был предложен механизм теоретико-множественных операций (ТМО) над ограниченным классом КСГ или предикативных грамматик, эквивалентных И-ИЛИ-дереву (т.е. - без рекурсии). Данный механизм имеет две модификации: с известной семантикой формального текста и – с неизвестной семантикой. Механизм ТМО с неизвестной семантикой основан на модификациях классических методов СА. Механизм ТМО с известной семантикой основан на обобщении

списка связей (цепей) в модели объекта. На вход ТМО поступает список прототипов, а результатом работы алгоритма будет и-или дерево, полученное путем их обобщения с атрибутами, определяющими принадлежность и-или узла к тому или иному прототипу. Поскольку VHDL, как и другие языки программирования, можно описать при помощи КСГ, данный метод применим к решаемой задаче.

ТМО над грамматиками так же использовались в работе [15] авторами системы ИСИДА для наполнения ресурса знаний в системе извлечения информации из текста. Однако, в этом случае, знания представлялись в виде графа и работа осуществлялась над ЕЯ-текстами.

Выбор метода извлечения знаний

В разрабатываемой интеллектуальной надстройке над САПР VHDL предполагается автоматический режим создания базы знаний, т.е. отсутствие инженера по знаниям. Задача будет рассматриваться в режиме «глупого эксперта», т.е. участие эксперта предметной области в процессе создания базы знаний будет минимальным [16]. В качестве источника знаний выступают библиотеки программ на языке VHDL, классификация которых приведена в работе [1]:

- написанные вручную (пользовательские программы);
- перекодированные (полученные путем автоматической перекодировки в язык VHDL);
- подготовленные (предоставляемые в стандартных библиотеках);
- сгенерированные (генераторы программ).

В случае если источниками VHDL-программ являются стандартные библиотеки, то при решении задачи извлечения знаний отпадает необходимость первоначального этапа разбиения готовых решений (VHDL-программ) на классы. Зная, структуру стандартной библиотеки легко можно определить к какому классу относится устройство. В другом случае, если источники знаний (VHDL-программы) являются личным опытом проектировщика или используются ресурсы интернет, эта задача остается актуальной, так как заранее неизвестно, к какому классу устройств относится тот или иной файл (VHDL-решение). Необходимо также учитывать, что VHDL-программы имеют, с одной стороны,

четкую структуру и соответствуют грамматике языка VHDL, а, с другой стороны, содержат субъективные данные (имена переменных, название модулей, комментарии), отражающие методики проектирования данного конкретного пользователя. С целью учета субъективной составляющей для наиболее полного отражения методики проектирования разработчика при анализе VHDL-текста возможно применение к нему некоторых методов используемых при анализе ЕЯ-текстов. Например, использование механизма создания словарей, используемых имен переменных, устройств, процедур с созданием синонимичных словарей для каждого термина так, как это описано в работе [10].

В качестве метода извлечения знаний предлагается использовать механизм ТМО над грамматиками, который формирует и-или деревья прототипов решений, чем обеспечивает наглядность модулей знаний и удобство их обработки. А использование атрибутов в и-или узлах дерева обеспечит возможность сохранения семантики рассматриваемого решения.

Важной задачей в процессе создания БЗ является оценка эффективности разрабатываемых методов. Эту задачу можно решить при помощи таких критериев, как точность метода извлечения знаний, полнота базы знаний и др.

Заключение

В работе проведен анализ методов извлечения знаний, отмечены их достоинства и недостатки, описаны основные направления извлечения знаний из ЕЯ-текстов. Рассмотрены основные инструменты анализа формальных текстов.

Для извлечения знаний из VHDL-текстов предполагается использовать метод, основанный на ТМО над формальными грамматиками, частично эквивалентный методам деревьев решений. Учет субъективной составляющей VHDL-решений разработчика будем осуществлять при помощи создания и ведения синонимичного словаря имен входов/выходов, устройств и т.д.

Следующий шаг работы построение базы знаний интеллектуальной надстройки над САПР VHDL и разработка метода извлечения знаний из VHDL-файлов с учетом описанных требований.

Список литературы

1. Григорьев А. В. Анализ средств автоматизации построения VHDL-программ / А.В. Григорьев, Д.А. Грищенко // Наукові праці ДонНТУ випуск 14(188) Серія "Інформатика, кібернетика та обчислювальна техніка. – 2011. – С. 262-269.
2. Гаврилова Т.А. Активные индивидуальные методы извлечения знаний и данных/ Т.А. Гаврилова // Enterprise Partner. Корпоративные системы. – 2001. – №18 (35). – С. 6–9.
3. Гаврилова Т.А. Извлечение знаний: «пассивные» методы / Т.А. Гаврилова // Enterprise Partner. Корпоративные системы. – 2001. – №11(28).

4. Арсеньев С. Извлечение знаний из медицинских баз данных [Электронный ресурс] / С. Арсеньев. – 1999. – Режим доступа: [http://neural.narod.ru/Arsen.html]
5. Вагин В.Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В.Н. Вагин, Е.Ю. Головина, А.А. Загорянская, М.В. Фомина. — М.: ФИЗМАТЛИТ, 2004. — 704 с.
6. Васяева Т.А. Извлечение знаний на основе генетических алгоритмов и генетического программирования / Т.А. Васяева, Ю.А. Скобцов // ВІСНИК Східноукраїнського національного університету ім. Даля. – 2008. – Ч. 2, №12 (130). – С. 5–11.
7. Han J. Data Mining: Concepts and Techniques / Jiawei Han, Micheline Kamber. – Morgan Kaufmann, 2006. – P. 770
8. Скатов Д.С. Модели и методы анализа иерархически структурированных текстов / Д.С. Скатов, Т.Н. Ерехинская, В.В. Окатьев // Материалы международной конференции по компьютерной лингвистике [«Диалог 2009»]. – М, 2009. – С. 346 – 351.
9. Ильин Н. Технологии извлечения знаний из текста / Н. Ильин, С. Киселёв, С. Танков, В. Рябышкин // «Открытые системы». – 2006. – №6. – С. 51–56.
10. Ермаков А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста / А.Е. Ермаков // Материалы международной конференции [«Диалог 2008»]. – М, 2008. – С. 154–159.
11. Иванисенко В.А. Извлечение знаний из текстов научных публикаций и создание баз знаний в области нанобиотехнологии / [В.А. Иванисенко, Н.Л. Подколотный, и др.] // Российские нанотехнологии. – 2011. – Т. 6, № 7/8. — С. 14–21.
12. Симаков К.В. Модель извлечения знаний из естественно-языковых текстов / К.В. Симаков, А.М. Андреев, Д.В. Березкин // Информационные технологии. – 2007. – №12. – С. 57–63.
13. Luboš Lorenc. A Note on the Parsing of Complete VHDL-2002 [Электронный ресурс] // Lorenc Luboš, Schonecker Rudoif, Křivka Zbynek. – Режим доступа: [http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-255/paper11.pdf]
14. Григорьев А.В. Специфика выполнения теоретико-множественных операций над контекстно-свободными грамматиками в условиях различных форм дополнительных семантических правил в семиотической модели интеллектуальных САПР / А.В. Григорьев // Научные труды ДонНТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП – 2006). – 2006. – Выпуск 5 (116). – С. 91-104.
15. Александровский Д.А. Реализация ресурса знаний в системе извлечения информации из текста / [Д.А. Александровский, Д.А. Кормалев, Е.П. Куршев, Е.А. и др.] // Материалы Одиннадцатой национальной конференции по искусственному интеллекту с международным участием (КИИ-08). – Дубна, 2008.
16. Григорьев А.В. Пути создания интеллектуальных САПР при различных уровнях квалификации экспертов / А.В. Григорьев // Научно - теоретический журнал «Искусственный интеллект». – 2005. – №3. – С. 758–763.

Надійшла до редакції 30.09.2012

О.В. ГРИГОР'ЄВ, Д.О. ГРИЩЕНКО
Донецький національний технічний університет

A.V. GRIGORIYEV, D.A. GRISHCHENKO
Donetsk National Technical University

Аналіз методів розв'язання задачі вилучення знань про методики проектування з VHDL-текстів.

An Analysis of the Methods of Extracting Knowledge about Design Techniques from VHDL Texts.

В статті відзначається актуальність вилучення знань про методики проектування на мові VHDL, розглянуті існуючі методи вилучення знань та можливі шляхи вилучення знань про методики проектування в інтелектуальній надстройці над САПР VHDL.

The paper discusses the importance of design techniques data extraction for VHDL language. It considers the existing methods of knowledge extraction and possible ways of extracting knowledge about the design techniques in intelligent VHDL CAD add-on.

Ключові слова: VHDL, САПР, інтелектуальна надбудова, методи вилучення знань, лексичний аналіз, синтаксичний аналіз, природньо-мовні тексти, формальні мови

Keywords: VHDL, CAD, intelligent VHDL CAD add-on, knowledge extraction methods, lexical analysis, parser, natural-language texts, formal languages