

УДК 004.822

А.В. Григорьев, канд. техн. наук, доцент,
Е.В. Павловский, аспирант,
Донецкий национальный технический университет, г. Донецк, Украина
grigorie@pmi.dgtu.donetsk.ua, pavlovskiy.evgeny@mail.ru

Подход к построению интеллектуальных CMS

В статье рассмотрены достоинства и недостатки существующих систем для построения web-приложений. Рассмотрена задача разработки инструментальной системы для построения интеллектуальных САПР web-приложений как экспертных систем, в базу знаний которой вводится методика проектирования сайтов. Рассмотрен предлагаемый механизм представления знаний.

Ключевые слова: CMS, онтология, САПР, база знаний, мета-эвристическая оболочка, ИЛИ-дерево, экспертная система, декомпозиция, эксперт предметной области

Введение

В настоящее время рынок фирм и организаций, которые предоставляют услуги в области web-технологий, растет с каждым днем. Это положительно сказывается на цене конечного продукта, т.к. конкуренция способствует удешевлению продукта или, по крайней мере, предоставляет клиенту диапазон цены продукта. Эта цифра содержит в себе заработную плату сотрудников, которые задействованы в разработке веб-приложения.

Актуальной задачей является оптимизации трудозатрат. Сокращая количество персонала, возникает проблема увеличения времени на разработку, а этот параметр, является одним из главных для заказчика. Второй момент заключается в том, что рядовой пользователь сейчас на своем ПК владеет очень большим выбором программных средств для решения тех или иных задач, таких как запись дисков, редактор текстов или графический редактор. У пользователя возникает вопрос необходимости такой системы, с помощью которой он мог бы самостоятельно создать веб-приложение, ориентированное на его уровень и удовлетворяющее его потребности.

Еще одна проблема, с которой сталкиваются web-разработчики – это повторное использование кода. Трудоемко писать каждое приложение «с нуля», тем более, что прототип подобного приложения уже есть. Таким образом, возникает необходимость внедрения в технологию разработки web-приложений системы автоматического проектирования (САПР) [1]. В настоящее время они представляют собой проблемно-ориентированный графический редактор, практически лишенный интеллектуальной поддержки методик проектирования. Таким образом, актуальной задачей является разработка инструментальной оболочки для

построения интеллектуальных САПР, в базу знаний которой вводится методика проектирования web-приложений.

Таким образом, мы сталкиваемся с задачей, которая касается специфики объекта проектирования. В последнее время разработчики пытаются наделить свои программные продукты интеллектуальностью. Это могут быть подсказки пользователю в ответ на его действия, интеллектуальное поведение без уведомления пользователя, которое базируется на ранее полученных данных от него (корректировка действий) и т.д. Программные продукты наделяются функциями, которые присущи экспертным системам [2]. Для данной предметной области существуют системы управления содержимым CMS (от англ. Content Management System) [3].

В настоящее время в области искусственного интеллекта существует тенденция замены инженера по знаниям, в процессе построения баз знаний, на эксперта предметной области. Это позволит эксперту создавать web-приложения самостоятельно, без привлечения инженера по знаниям. Применение такого подхода позволило бы эксперту создавать базу знаний по синтезу web-приложений самостоятельно, без привлечения инженера по знаниям. Желаемый САПР может быть построен либо как база знаний – надстройка над существующим CMS, либо – как полностью автономный интеллектуальная CMS. Построенная САПР позволил бы конечному пользователю синтезировать требуемую САПР по техническому заданию. Указанный подход реализован к концепции мета-эвристических оболочек (МЭО) для построения интеллектуальных САПР. Из трех возможных режимов работы МЭО в данном случае наиболее подходящим является режим «среднего» эксперта [4]. Такой эксперт в предметной области способен построить

онтологию предметной области [5,6] как совокупность модулей знаний, каждый из которых представляет собой комплекс И-ИЛИ-деревьев [7] с определенными над ним отношения несовместности, косвенно задающими продукции базы знаний.

Цель предлагаемой работы: используя концепцию МЭО, предложить путь создания интеллектуальных CMS, обеспечив возможность их построения эксперту в предметной области, имеющему достаточно большой опыт в разработке web-приложений.

Постановка задачи

1. Исследовать уровень интеллектуальности существующих CMS;
2. Предложить путь построения интеллектуальных CMS, для построения которых нет необходимости привлекать инженеров по знаниям, а можно обеспечить возможность ее построения эксперту в предметной области, имеющему достаточно большой опыт в разработке web-приложений.

Решение задач и результаты исследований

1. Представление объекта проектирования

Объектом проектирования САПР является web-приложение, или в частном случае сайт, который соответствует всем требованиям эксперта. В процессе синтеза решается задача выбора решения из ограниченного множества решений, удовлетворяющего пользователя. Нужно разработать инструмент для выбора наиболее подходящего решения. Система должна решить следующие задачи: 1) оптимизация трудозатрат; 2) сокращение времени на создание конечного продукта; 3) создание БЗ с использованием подходов, интуитивно понятных эксперту в предметной области, который не является экспертом в инженерии знаний; 4) разработка средств синтеза web-приложений в рамках готовой БЗ, организующих диалог на уровне, интуитивно понятном конечному пользователю, который может даже не являться экспертом в web-разработке; 5) синтез сайта на проблемно-независимом языке, на основании БЗ и ответов пользователя; 6) конвертация из проблемно-независимого языка в проблемно-зависимый (в частном случае – это язык HTML).

2. Обзор существующих САПР

На сегодняшний день существует ряд CMS для автоматического создания сайтов. Другими словами, это система управления

содержимым. Данные системы позволяют добавлять, редактировать и удалять содержимое на сайте. Манипуляции с содержимым проходят посредством административной части, встроенной в ядро системы. Система позволяет в удобном и понятном для пользователя интерфейсе совершать действия с информацией.

Информация может быть представлена в виде текста, картинки или видео. Информация о ней размещается в базе данных (БД) системы. Сложная реляционная БД позволяет информацию (материал) группировать по типу. Также система, обычно позволяет группировать материал в виде дерева, что позволяет придать четкую иерархию материала (таксономия) [8]. Например, тип материала «новости», материал «ДТП в Донецке». Если же информацию нужно как-то упорядочить и группировать, то создается, так называемая таксономия.

Под таксономией понимается словарь и вложенные в него термины. Вложенность терминов друг в друга не ограничена. Например, словарь носит название «Образование». У него есть термины «Предмет», в который вложены термины «физика» и «химия». В свою очередь химия делится на термодинамику и гидродинамику. Вид такой таксономии представлен на рис. 1.



Рисунок 1 – Таксономия словаря «Образование»

Таксономия позволяет четко структурировать информацию и придавать ей иерархическую вид. Это позволяет на одной web-странице отображать статьи, которые относятся к химии, а на другой к физике. Кроме того, родительская ветка, может включать в себя только свой материал или же дочерние ветки с указанной вложенностью.

Обычно в такую систему встроена возможность работы с пользователями системы. Существует возможность регистрации нового пользователя, редактирование его данных, а также удаление. Каждый пользователь может иметь одну или несколько ролей, такие как администратор или модератор. Это позволяет ограничивать доступ к информации по ролям. К административной части CMS есть доступ только у администратора сайта.

Система позволяет выбирать макет (тему) для внешнего вида web-страниц сайта. Этот

макет поделен на именованные области. Это нужно для размещения новых сущностей (модулей, страниц и т.д.). Т.е. видимые объекта на страницы связаны с именованной областью. Обычно в систему встроены базовые макеты, но пользователь может загрузить в систему любой понравившийся ему макет, разработанный для этой системы. По своей структуре, темы похожи друг на друга. Обычно в верхней части находится область для картинки (логотипа), которая обычно занимает всю ширину. Под или над ней находится область для главного меню. Затем, обычно ниже, находится область для основного содержания, новостей, копирайта и т.д.

С помощью настроек сайта можно изменить язык интерфейса, что позволяет пользоваться системой людям со знанием различных языков. Т.е. базовые интерфейсы системы переведены разработчиками на множество языков.

Все перечисленные функции позволяют создать полноценный сайт за короткое время. Обычно создание сайта можно разбить на следующие этапы:

1. Установка локального сервера с поддержкой PHP и MySQL требуемых версий на домашний ПК.

2. Установка CMS на локальный сервер, которая включает в себя регистрацию нового пользователя с правами администратора и настройку базовых параметров системы.

3. В административной части CMS создается материал, классифицируется для дальнейшего размещения на страницах. Один из материалов помечается как материал для главной страницы

4. Организовать ссылки для доступа к материалам (зачастую посредством меню).

Пользователь может использовать встроенный в систему редактор, который позволяет неявно разместить информация внутри html-тегов (контейнеров). Плюсы такого редактора заключаются в том, что пользователь работает только с обычным текстом. Редактор сам расставит нужные теги, на основании визуального редактирования пользователем.

Среди преимуществ CMS можно выделить следующие:

1. *Возможность размещать и редактировать материал*, пользователем любой квалификации. Для этого в системе существует понятный и доступный интерфейс для работы с информацией. Редактор сам заметит обычный текст с помощью html-тегов. Это предотвращает затраты на привлечения специалиста со стороны.

2. *Уменьшение стоимости и времени* на разработку сайта, что достигается использованием готовых модулей для системы.

Эти модули разрабатываются специально для конкретной системы. Они постоянно тестируются пользователями и обновляются по мере необходимости.

3. *Организация совместной работы*. Система позволяет работать над одной и той же информацией многим пользователям и ограничивать доступ к материалам в зависимости от прав пользователя. Администратор может редактировать роли пользователя, что положительно сказывается на работе предприятия, каждый отдел которого, имеет доступ только к своим данным.

4. *Безопасность системы*. Один из важнейших преимуществ CMS. В крупных компаниях, занимающихся разработкой сайтов, уделяется огромное внимание теме безопасности. Это касается как сохранение целостности и конфиденциальности данных, так и работоспособности сайта в целом.

На сегодняшний день существует достаточно большое количество CMS. Мы рассмотрим наиболее популярные из них. Для того, чтобы это сделать, нужно определить, на какие категории они разделены.

CMS можно разделить по следующим критериям:

1. Распространение:

- а) платные;
- б) бесплатные.

2. Освоение:

- а) легкие;
- б) требует определенных знаний;
- в) сложные.

3. Основное содержимое:

- а) блог;
- б) интернет-магазин;
- в) форум;
- в) универсальная.

4. Популярность (востребованность):

- а) редко используются;
- б) востребованы;
- в) популярные.

5. Нагрузка на сервера

- а) не значительная;
- б) значительная;
- в) большая.

6. Поддержка сообщества программистов:

- а) слабая;
- б) хорошая;
- в) отличная.

Рассмотрим самые популярные и известные CMS системы и дадим им сравнительную характеристику (см. табл. 1).

Таблица 1 - Сравнительная характеристика популярных CMS

Критерий \ CMS	Drupal	Joomla	WordPress	1С-Битрикс
Распространение	бесплатная	бесплатная	бесплатная	платная
Освоение	требует определенных знаний	легкое	требует определенных знаний	сложное
Основное содержимое	универсальная	универсальная	блог	универсальная
Популярность	популярная	популярная	популярные	популярные
Нагрузка на сервера	не значительная	не значительная	значительная	большая
Поддержка сообщества	отличная	отличная	хорошая	поддержка фирмой 1С
Достоинства	<ol style="list-style-type: none"> 1. Бесплатная CMS 2. Простота установки и интуитивно понятный интерфейс. 3. Простой и удобный API 4. Большой выбор доступных тем оформления и модулей. 5. Достаточно мощное русскоязычное сообщество программистов. 6. Русская документация. 7. Высокая стабильность. 	<ol style="list-style-type: none"> 1. Бесплатная CMS 2. Большой выбор доступных тем оформления и модулей. 3. Управление системой не требует наличия специальных знаний и навыков. 	<ol style="list-style-type: none"> 1. Бесплатная CMS 2. Большой выбор доступных тем оформления и модулей. 3. Высокая популярность гарантирует помощь со стороны сообщества. 	<ol style="list-style-type: none"> 1. Высокая надежность и безопасность системы. 2. Техническая поддержка, т.к. система платная. 3. Подробная документация.
Недостатки	<ol style="list-style-type: none"> 1. Слабая реализация принципов ООП при разработке ядра системы. 2. Достаточно сложная в освоении система, особенно для рядового пользователя. 	<ol style="list-style-type: none"> 1. Слабая безопасность от взлома. 2. Скорость загрузки web-страниц. 	<ol style="list-style-type: none"> 1. Ограниченная расширяемость. 2. Проблемы с безопасностью. 3. Существенная нагрузка на сервер при большой посещаемости. 	<ol style="list-style-type: none"> 1. Платная CMS. 2. Существенно нагружает сервер. 3. Сложность системы требует от пользователя знания PHP.

База знаний (БЗ) таких систем отсутствует. Вместо нее профессионалами в области web-программирования разрабатываются модули. Они из себя представляют код на формально-зависимом языке, что ограничивает пользователя средней квалификации в изменении существующих модулей и добавление новых.

Типичная CMS представляет собой проблемно-ориентированный графический редактор, практически лишенный интеллектуальных средств поддержки процесса проектирования web-приложений. Для построения базы знаний мы предлагаем привлекать не инженера по знаниям, а эксперта в предметной области, имеющего достаточно большой опыт разработки web-приложений.

3. Представление знаний

Таким образом, возникает задача разработки инструментальной системы для построения интеллектуальных САПР. Т.е. инструментальная система строит САПР как экспертную систему, в базу знаний которой

вводится методика проектирования сайтов. Он позволяет конечному пользователю

синтезировать требуемый сайт по техническому заданию, которое с точки зрения продукционной базы знаний [9], рассматривается как «цель вывода». В процессе синтеза, т.е. логического вывода, по заданной цели вывода в продукционной базе знаний строится результат вывода, т.е. готовый сайт.

В режиме диалога с пользователем исключаются некоторые варианты. до тех пор, пока не останется других вариантов. Также для ускорения процесса интервью, пользователь может задать связь несовместности между понятиями. Таким образом, если два понятия А и В связаны отношением несовместности, то если А нужно на сайте, то все ветки дерева с понятием В удаляются.

С введением таких диаграмм, система приобретает интеллектуальность, чего не хватает в существующих CMS, так как они рассчитаны на квалифицированных пользователей. Таким образом, можно решить проблему проектирования сайтов пользователем среднего уровня квалификации.

Выбор структуры данных для представления знаний – важнейший этап проектирования. Структура должна быть понятна эксперту и при этом должна минимизировать временные и технические затраты на обработку запроса.

Таким запросом может быть:

- навигация;
- быстрый поиск нужного понятия;
- добавление и редактирование понятия;
- перемещение понятия с подчиненными понятиями;
- удаление понятия с подчиненными понятиями.

Как мы видим, имеется достаточно много требований к структуре. В искусственном интеллекте основная цель – хранить знания

таким образом, чтобы программы могли извлекать их и работать с ними. САПР должен решить задачу перевода графической информации, полученной в результате проектирования БЗ с помощью графических примитивов, в структуру данных для последующей обработки. Нужно решить задачу выбора подходящей модели представления знаний.

Имеются следующие модели знаний:

- логические;
- сетевые;
- продукционные;
- фреймовые.

Проведем сравнительную характеристику каждого метода (см. табл. 2).

Таблица 2 - Сравнительная характеристика моделей представления знаний

	Логическая модель	Сетевая модель	Продукционная модель	Фреймовая модель
Характеристика	Представление знаний в виде формальных систем (теорий). Используется язык логики предикатов.	Выделение некоторых фиксированных множеств объектов и задание различного рода отношений на них.	Основывается на множестве правил вида «Если (условие), то (действие)».	Фиксированная жесткая структура информационных единиц В общем виде фрейм имеет вид: [Имя фрейма:] [Имя слота 1: значение слота 1] [Имя слота 2 (значение слота 2)] [и т.д.]
Достоинства	Возможностями исчисления предикатов, теоретической обоснованностью выводов, осуществляемых в системе.	Сравнительно малое время автоматизированного поиска информации, а с другой стороны. Удобство при описании определенных областей знаний	Эффективная модель по затратам памяти и по быстродействию. Обладает наглядностью, высокой модульностью, простотой механизма логического вывода, легкостью внесения дополнений и изменений.	Сравнительно малое время автоматизированного поиска информации, а с другой стороны. Удобство при описании определенных областей знаний
Недостатки	Значительные проблемы с организацией вывода	Сложность процедуры поиска вывода на семантической сети	Противоречивость продукции при их большом количестве	Значительные проблемы с организацией вывода

Одной из самых сложных задач САПР является обеспечение механизма логического вывода. Проанализировав характеристики, достоинства и недостатки всех вышеперечисленных моделей представления знаний, приходим к выводу, что продукционная модель является наилучшим вариантом из рассмотренных. Также стоит взять во внимание опыт разработки уже существующих систем, таких как CLIPS [10] и OpenCus, которые основаны на продукционной модели представления знаний.

Но эксперт, в силу своей квалификации, не может строить продукции явным образом. Поэтому не одна из этих моделей не подходит (см. табл. 1). Таким образом самой оптимальной структурой данных является онтология, которая представляет из себя декомпозицию понятий [5].

Она классифицирует понятия предметной области, задает связь между ними и ограничения. В нашем случае требуемая онтология представляет собой И-ИЛИ-дерево с определенным над ним отношениями несовместности.

Инструментальная оболочка позволяет эксперту в предметной области, выполняющему роль инженера по знаниям, построить И-ИЛИ дерево со всеми возможными вариантами связей и атрибутов.

Процесс построения базы знаний разбит на последовательный ввод модулей знаний, от более крупных - к более мелким. При синтезе движение по модулям знаний имеет тот же порядок – от более крупных к более мелким. С помощью готовой базы знаний можно выбрать модуль знаний любого уровня и запустить логический вывод.

Модуль знань розбивається на підмодулі [4, 6]:

1) Внешняя граница как состав внешних параметров, данных, которые отражают связи блока с внешней средой. Это отдельное И-ИЛИ-дерево с отношениями несовместности;

2) Внутренняя граница как состав внутренних подблоков, тоже как И-ИЛИ-дерево;

3) Совокупность связей между подблоками по данным, тоже как И-ИЛИ-дерево.

В рамках любого И-ИЛИ-дерева определяются отношения несовместности – как связь, которая показывает взаимоисключающие элементы в И-ИЛИ-дереве. Т.о., если в дальнейшем на этапе диалога с пользователем выясняется, что один элемент будет присутствовать на странице, то автоматически второй удаляется из И-ИЛИ дерева. Это может существенно сократить время опроса пользователя и сузить количество вариантов решений. Кроме того, смежные подмодули тоже связаны отношениями несовместности.

В процессе вывода в рамках подмодуля знань система вывода движется в некотором порядке по всем ИЛИ-узлам. Выбрав какой-то ИЛИ-узел, система вывода делает из него вопрос, предлагая ряд вариантов ответов. Например, пользователь выбирает один из вариантов ответа на вопрос: «Имеет ли место блок В?». После ответа система по отношениям несовместности в ИЛИ-дереве удаляет в других ИЛИ-узлах те альтернативы, которые несовместимы с наличием нужного блока. Таким образом, построенный САПР как экспертная система позволяет конечному пользователю синтезировать требуемый САПР по техническому заданию.

Достоинства предлагаемого подхода – это возможность разработки конечным пользователем типичных сайтов без использования кода прототипа. При этом допускается создания сайтов различных классов.

4. Пример реализации подхода

Инженером по знаниям является эксперт предметной области средней квалификации, который не может описать продукции явным образом, но может построить И-ИЛИ дерево и задать отношение несовместности, таким образом описать продукции неявно.

Выделим в типичном сайте следующие блоки:

1. Системообразующие.
 - заголовок;
 - логотип;
 - меню;
 - основное содержание (контент).

2. Факультативные.

- Поиск;
- Обратная связь;
- Голосование;
- Авторизация;
- Блог;
- Форум.

Предполагается, что данные понятия предметной области изначально известны системе и они связаны с соответствующими подпрограммами. На примере приведенных выше блоков, построим И-ИЛИ дерево (см. рис 2). Здесь имеются «ИЛИ» узлы – варианты декомпозиции блоков «Контент» и «Авторизация»

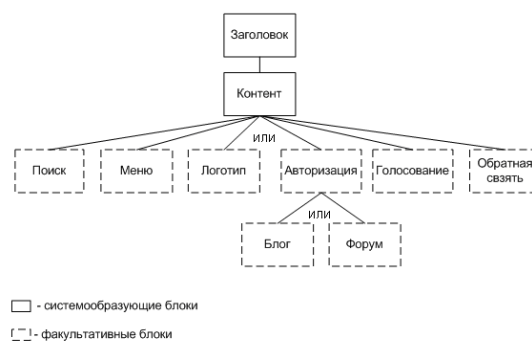


Рисунок 2 – БЗ в виде И-ИЛИ дерева (первый уровень)

На рис. 2 представлен первый уровень БЗ. Исходя из специфики предметной области, на первом уровне БЗ не можем выявить связи несовместности явно. В случае с блоком авторизации, из И-ИЛИ дерева четко вытекает связь совместности блоков «Блог» и «Форум» с блоком «Авторизация». Из этого следует, что подблоки являются совместными с блоком и не требуют явным образом указания связи совместности. Таким образом, отсутствие авторизации на сайте, повлечет за собой невозможность присутствия в нем блога и форума.

Декомпозируем по «И» блок «Авторизация» и таким образом перейдем к уровню 2 модели Мира (см. рис. 3).

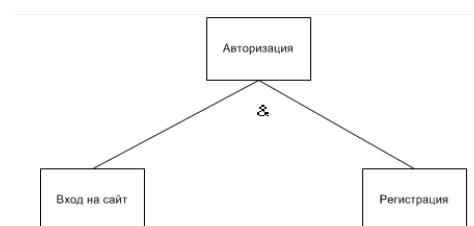


Рисунок 3 – декомпозиция блока «Авторизация» (второй уровень)

Далее декомпозируем подблоки блока «Авторизация» (см. рис. 4). Блок «Капча» – это блок, связанный с модулем, который предоставляет картинку с надписью. Эту надпись нужно ввести в соответствующее поле для ввода в форме регистрации. Этот механизм позволяет предотвратить регистрацию на сайте не человека.

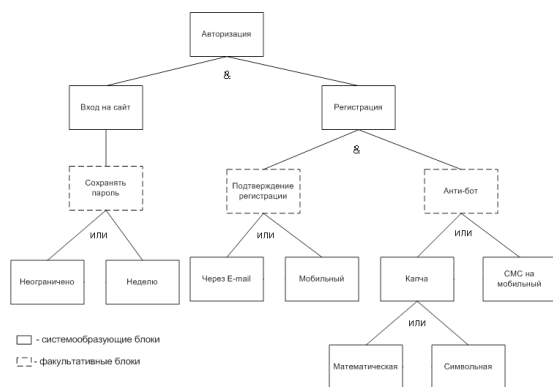


Рисунок 4 – Декомпозиция блока «Вход на сайт» и «Регистрация» (третий уровень)

Далее декомпозируем блок «Капча» (см. рис. 5).

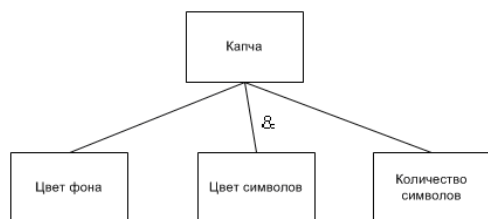


Рисунок 5 – Декомпозиция блока «Капча» (четвертый уровень)

Процесс построения базы знаний разбит на последовательный ввод модулей знаний, от более крупных к более мелким. При синтезе с помощью готовой базы знаний можно выбрать модуль знаний любого уровня и запустить логический вывод.

Модуль знаний разбивается на подмодули [11]:

- внешняя граница как состав внешних параметров, данных, которые отражают связи блока с внешней средой. Это отдельное И-ИЛИ-дерево с отношениями несовместности;
- внутренняя граница как состав внутренних подблоков, тоже как И-ИЛИ-дерево;
- совокупность связей между подблоками по данным, тоже как И-ИЛИ-дерево;

Кроме того, смежные подмодули тоже связаны отношениями несовместности. В

процессе вывода в рамках подмодуля знаний система вывода движется в некотором порядке по всем ИЛИ-узлам. Выбрав какой-то ИЛИ-узел, система вывода делает из него вопрос типа:

Есть особенность в составе блоков. Какой вариант Вам нужен:

- 1) С блоком авторизации;
- 2) Без блока авторизации.

Например, пользователь выбирает вариант 1. Тогда система по отношениям несовместности в ИЛИ-дереве удаляет в других ИЛИ-узлах те альтернативы, которые несовместимы с наличием блока авторизации.

Дерево сужается, но внутренняя структура модулей не сужается. Диалог идет дальше, пока не останется ИЛИ-узлов. Затем переходим к нижележащему подмодулю и т.д.

На выходе – готовая структура блока, включая [11]:

- внешнюю границу;
- состав подблоков;
- состав связей между ними.

Любой подблок, если он не базовый, это тоже вызов конкретного модуля знаний. Только у него уже сформировался конкретный набор входов-выходов на внешней границе по списку связей. Т.е., фактически этап синтеза внешней границы уже состоялся. Осталось перейти к подблоком, а затем – к связям.

Далее процесс повторяется.

Если все блоки базовые – процесс синтеза останавливается.

Более детально рассмотрим И-ИЛИ-дерево разных уровней. Из рис. 2 видно, что в дереве есть системообразующие блоки и факультативные. Системообразующие блоки – это неотъемлемая часть системы, поэтому вопрос нужен он или нет, пользователю не задается. Установив в режиме диалога блоки, которые будут размещены на сайте, пользователю предлагается ответить на вопросы, которые касаются подблоков каждого блока (декомпозиция блоков).

На примере блока «авторизация» (см. рис. 3), можно видеть, что этот блок состоит из двух системообразующих блоков. Значит, на этом уровне вопросы к пользователю отсутствуют. Декомпозируем каждый из этих блоков (см. рис. 4).

При декомпозиции блока «Регистрация» (см. рис. 4), имеем более сложную структуру. Блоки «Подтверждение регистрации» и «Анти-бот» являются факультативными. Последний уровень декомпозиции – это детализация на уровне атрибутов. На примере блока «Капча» (см. рис. 5), можно видеть атрибуты, присущие этому блоку. Все они системообразующие.

5. Структурный уровень САПР

Как видно из предыдущей главы, эксперт представляет БЗ в виде онтологии, причем многоуровневой. На верхнем уровне находятся модули знаний, а на последнем – атрибуты. Это своего рода диаграммы IDEF0 [10]. Онтология может иметь много уровней детализации, что обеспечивает бесконечную вложенность блоков.

Далее следует сказать о взаиморасположении блоков на странице сайта. Изначально пользователь должен выбрать макет из предложенного перечня или же задать новый вид. Макет представляет из себя матрицу. Каждая ее ячейка – это именованная область, которая может включать в себя блоки верхнего уровня.

На самом нижнем уровне находится макет, так называемый фундамент для построения полноценной БЗ. Это некий аналог печатной платы, используемой для проектирования материнских плат. Далее в определенные именованные области макета отношением блок – подблок связываются понятия. Понятие в нашем случае – это базовая единица предметной области. Его внутренняя структура и поведение инкапсулирована внутри него. Эксперт в силу своей квалификации не вникает в подробности его реализации. Если проводить аналогию с материнской платой, то понятие – это электронный компонент (например, микросхема). Эксперт знает, что она должна быть, но как она устроена внутри он не знает, но знает, каким отношением связаны макет и примитив. Так как САПР интеллектуальный, то эксперт может не задавать данную связь. Интеллектуальный механизм, который на этапе синтеза найдет элемент, который не входит в состав всей концептуальной модели, и предложит пользователю возможные варианты. Можно провести аналогию с электротехникой. Это случай, когда цепь разорвана и элементы цепи не функционируют.

Атрибут – это неотъемлемый параметр блока нижнего уровня. Он может быть

факультативный (не обязательный) или такой, без которого существование примитива не имеет смысла (рефлексивная связь) [11]. В проектировании материнских плат, атрибуты (параметры) ее компонентов играют огромную роль. Если взять микросхему, то к таким параметрам относятся: уровень энергопотребления, пропускная способность шины и т.д.

Представление знаний в виде онтологии позволяет наследовать атрибуты, что позволяет сократить время создания БЗ и обеспечит понятный и наглядный интерфейс наследования. Этот механизм позволяет переопределять атрибуты родителя, что не ограничивает пользователя в свободе действий.

Заключение

Были рассмотрены функциональные возможности типичных CMS, как САПР, решающих задачи построения web-приложений. Самый главный недостаток всех подобных систем – это ориентация на пользователя, который знаком с внутренним устройством сайтов и языком html, что не позволяет среднестатистическому пользователю создать сайт, который бы удовлетворял все его требования.

В предлагаемой статье:

- предложен подход к созданию интеллектуальных CMS, для построения которых нет необходимости привлекать инженеров по знаниям, а можно обеспечить возможность ее построения эксперту в предметной области, имеющему достаточно большой опыт в разработке web-приложений.

- в общем виде разобран механизм представления знаний в предлагаемой оболочке.

Полученные результаты дают возможность перейти к следующему этапу проектирования экспертной системы, а именно - к рассмотрению методов извлечения знаний из эксперта.

Список литературы

1. Ли К. Основы САПР (CAD/ CAM/CAE) / К. Ли. – СПб.: Питер, 2004. – 560 с.
2. Джаратано Джозеф. Экспертные системы: принципы разработки и программирование / Джозеф Джаратано, Гари Райли; пер. с англ. – [4-е изд.]. - М.: Издательский дом «Вильямс», 2007. – 1152 с.
3. Горнаков С. Г. Осваиваем популярные системы управления сайтом (CMS) / С. Горнаков. – М.: ДМК Пресс, 2009. – 336 с.
4. 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.
5. Григорьев А.В. Анализ методов построения онтологий для построения экспертных систем по синтезу моделей сложных систем в САПР / А.В. Григорьев, Е.В. Павловский // Научные труды

Донецкого государственного университета Серия: "Вычислительная техника и автоматизация". – 2011. – № 21. – С. 87-94.

6. Григорьев А.В. Комплекс моделей САПР как система взаимосвязанных уровней знаний о действительности / А.В. Григорьев // Научные труды Донецкого государственного университета: Серия «Информатика, кибернетика и вычислительная техника». – 2000. – Вып. 10. – С. 155-167.

7. Григорьев А.В. И/ИЛИ-дерево как средство абстрактного представления знаний / А.В. Григорьев, А.А. Каспаров // Наукові праці національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка». – 2002. – Вып. 39. – С. 36-42.

8. Мерсер Д. Dgural 6. Создание надежных и полнофункциональных веб-сайтов, блогов, форумов, порталов и сайтов-сообществ / Д. Мерсер. – М.: Вильямс, 2009. – 272 с.

9. Братко И. Программирование на языке Пролог для искусственного интеллекта / И. Братко; пер. с англ. – М.: Мир, 1990. – 560 с.

10. Частиков А.П. Разработка экспертных систем. Среда CLIPS / А.П. Частиков, Т.А. Гаврилова, Д.Л. Белов. – СПб.: БХВ-Петербург, 2003. – 608 с.

11. Григорьев А.В. Семиотическая модель базы знаний САПР / А. В. Григорьев. // Научные труды Донецкого государственного технического университета Серия: «Проблемы моделирования и автоматизации проектирования динамических систем». – 1999. – №10 — С. 30-37.

12. Функциональное моделирование на базе стандарта IDEF0: Учебный курс. — Минск: Software development & business consulting, 2002. — 35 с.

Надійшла до редакції 13.04.2012

О.В. ГРИГОР'ЄВ, Є.В. ПАВЛОВСЬКИЙ
Донецький національний технічний університет

O.V. GRIGORIEV, Ye.V. PAVLOVSKY
Donetsk National Technical University

Підхід до побудови інтелектуальних CMS

An Approach to Building Intelligent CMS

У статті розглянуті переваги і недоліки існуючих систем для побудови веб-додатків. Розглянуто задачу розробки інструментальної системи для побудови інтелектуальних САПР веб-додатків, як експертних систем, в базу знань яких вводиться методика проектування сайтів. Розглянуто запропонований механізм представлення знань.

The paper discusses the advantages and disadvantages of existing systems for building web-applications. We consider the task of developing an instrumental system for building web-intelligent CAD web-applications as expert systems, into the knowledge base of which sites design methods are entered. The proposed mechanism of knowledge representation is considered.

Ключові слова: CMS, онтологія, САПР, база знань, мета-евристична оболонка, І-АБО-дерево, експертна система, декомпозиція, експерт предметної області

Keywords: CMS, ontology, CAD, database knowledge, meta-heuristic shell, AND-OR-tree, expert system, decomposition, domain expert