

АЛГЕБРО-АЛГОРИТМИЧЕСКИЕ СРЕДСТВА ПРОЕКТИРОВАНИЯ ЗНАНИЙ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

Ключевые слова: клоны, экология программирования, САА-схемы, модели.

ВВЕДЕНИЕ

Алгебраизация программирования как направление формализует опыт программирования и предоставляет алгебраические инструменты для его развития. В Украине такой подход стал развиваться с 1965 г. и воплощен в теории алгоритмики, основы которой заложены в публикациях ученых под руководством Екатерины Логвиновны Ющенко. Системы алгоритмических алгебр (САА) [1] формализовали проектирование и синтез (сборку) алгоритмов и программ. Эти объекты проектировались в терминах регулярных схем как алгебраических представлений в САА. Основными методами проектирования регулярных схем признаны интерпретация (развертка), абстрагирование (свертка), переинтерпретация (свертка с последующей разверткой) и трансформация. Развитие аппарата эквивалентных преобразований, что составляет основу трансформации, обеспечило оптимизацию объектов проектирования по выбранным критериям (например, память, быстродействие), определило пути перехода от одной предметной области к другой, близкой по алгоритмическим аспектам обработки.

Следствием развития САА послужила теория клонов [2]. Теория клонов формализует современные подходы к программированию: представление алгоритмических знаний различных предметных областей (ПО), формирование набора алгоритмических операций, наиболее адекватно отображающих специфику ПО, обеспечение функциональной полноты для такого набора. Объект проектирования, исследования и преобразования в теории клонов — спецификация проектируемой программы, представленная как алгебраическая абстракция. В [3] в качестве программной спецификации предложены абстракции, не зависящие от целевого языка, в среде которого синтезируется программа. Но в этих работах понятие абстракции недостаточно формализовано, в большой степени зависит от среды проектирования и не предусматривает преобразования синтезированных программ. В теории клонов разработан аппарат алгебраического представления абстракций, их преобразования и оптимизации. Предложены три взаимосвязанные формы их описания — аналитическая (алгебраическое выражение), лингвистическая (САА-схема) и визуальная (граф-схема). Переход от одной формы к другой осуществляется автоматически. Эти формы описания в совокупности с исследованием алгоритмических свойств системообразующих операций в клонах и проблемы их функциональной полноты определяют биологию программирования теории клонов. Экология программирования в теории клонов определена как инструментальная поддержка создания и сопровождения баз алгоритмических знаний ПО, состоящих из абстракций алгоритмов и базовых программных реализаций типовых операций предметной области.

Изложение материала статьи подчинено следующей структуре.

- Основные аспекты теории клонов. Клон n -отношений.
- Учетные модели и клон n -отношений.
- Инструментарий автоматизированного проектирования компьютерных приложений.
- Виртуальная аудитория — система, созданная с использованием инструментов теории клонов и автоматизирующая функционирование стационарных и дистанционных форм обучения в учебных заведениях. Система обобщает опыт создания web-приложений с использованием теории клонов. В ней заложены средства создания web-приложений, доступных пользователям с проблемами зрения. Эти инструменты можно использовать для создания web-сервисов, поскольку в них предложен подход для описания функциональной структуры web-сайтов с автоматизацией синтеза таких сервисов.

© Г.Е. Цейтлин, Л.М. Захария, 2009

ОСНОВНЫЕ АСПЕКТЫ ТЕОРИИ КЛОНОВ. КЛОН n -ОТНОШЕНИЙ

Понятие клона относится к числу фундаментальных понятий универсальной и общей алгебры [4, 5]. В алгебре алгоритмики клоны используют в качестве инструмента описания различных средств проектирования алгоритмических знаний в предметных областях с учетом специфики алгоритмических операций и структур данных. Под клоном понимают одно- или многоосновную алгебру

$$C ::= \langle F/q; \text{SUPER} \rangle, \quad (1)$$

где F/q — совокупность основ, $q=1,2,\dots,n$, каждая из которых состоит из множества функций определенного типа (логического, операторного, структур памяти и данных); SUPER — сигнатура клона, представляющая операцию суперпозиции функций, принадлежащих основам. В одну (головную) функцию вместо переменных определенных типов подставляются другие функции соответствующего типа. Каждая основа клона состоит из множества функций одного и того же типа.

Клон Поста (КП) — одноосновный клон; его основу представляет множество всех булевых функций (БФ). КП ориентирован на построение семейства алгебр БФ. Необходимость в построении такого семейства связана с традиционными приложениями двухзначной алгебры логики (так обычно называется КП) — конструированием комбинационных схем компьютерной аппаратуры, алгоритмов и программ в языках программирования (ЯП).

Представительной назовем элементарную полугрупповую алгебру A (с операцией $*$), по которой строится клон.

Перечисленные далее клоны формализуют проектирование алгоритмической структуры программных приложений. Рассмотрим пары: представительная алгебра и соответствующий ей алгоритмический клон [2]. Отметим, что каждой паре соответствует определенный метод и технология программирования.

— Структурное программирование: алгебра Дейкстры (АД) — клон Дейкстры (КД), системы алгоритмических алгебр (САА) Глушкова — клон Глушкова (КГ).

— Неструктурное программирование: алгебра Янова (АЯ) — клон Янова (КЯ).

— Визуальное программирование: алгебра граф-схем Калужнина (АК) — клон Калужнина (КК) (обобщенные АК — КК).

Каждая представительная алгебра является двухосновной и характеризуется логической и операторной компонентами. Логическая основа состоит из булевых операций. В клоне Глушкова логическая компонента дополнена операцией прогнозирования, проверяющей логические условия на ожидаемом состоянии памяти после предполагаемого выполнения оператора. С введением операции прогнозирования в клон Глушкова возрастает его эффективность при проектировании. Клоны Дейкстры, Калужнина и Янова в качестве логической компоненты используют клон Поста. Проблема функциональной полноты для этих клонов решена в [2] с учетом операции прогнозирования в КГ, что и объясняет изобразительную мощимость САА и клона КГ в сравнении с другими клонами.

Рассмотрим алгебру Клини $AK = \langle RS; \text{СИГН} \rangle$, где RS — основа, представляющая множество всех регулярных событий (РЕС) над алфавитом A ; СИГН — сигнатура, состоящая из следующих операций: умножения $x_1 * x_2$, объединения $x_1 \cup x_2$, итераций

$$IT(*) = \{x^*\} = \cup x^k \mid k=0, 1, 2, \dots; \quad IT(+) = \{x^+\} = \cup x^k \mid k=1, 2, \dots$$

Рассмотрим одноосновный клон $KL/A ::= \langle \text{ОП}(KL/A); \text{СУПЕР} \rangle$, где $\text{ОП}(KL/A)$ — основа, которую составляет совокупность всех возможных операций, определенных на KL/A ; СУПЕР — сигнатура, включающая только суперпозицию функций с возможностью отождествления и переименования их переменных. Если $\text{ОП}(KL/A)$ совпадает с сигнатурой алгебры Клини, то получаем клон Клини.

Отметим, что рядом с одноосновными клонами построены их многоосновные обобщения, в которых системы образующих содержат контекстно-свободные операции. Согласно [11] клон Клини имеет такие свойства: принадлежит к алгебрам континуального типа, охватывает конечно-порожденные подалгебры с бесконечным базисом и без базиса, при этом описаны все его максимальные подалгебры, до-

казана теорема о функциональной полноте. Как следствие, клоны формальных языков адекватны по свойствам k -значным логикам ($k > 2$). Клон Клини и исследование проблемы его функциональной полноты важно не только в контексте теории формальных языков, но и в его связи с клоном n -отношений.

Для клона n -отношений определим соответствующую ему трехосновную представительную алгебру. Двумя основами такой алгебры являются известные логическая и операторная компоненты систем алгоритмических алгебр Глушкова. В качестве третьей компоненты выберем алгебру n -отношений. Определим основное понятие такой алгебры — понятие n -арного отношения.

Под операцией $M_1 \times M_2 \times \dots \times M_n$ декартового произведения множеств M_1, M_2, \dots, M_n будем понимать совокупность всех упорядоченных n -ок (a_1, a_2, \dots, a_n) , $a_1 \in M_1, a_2 \in M_2, \dots, a_n \in M_n$. В частности, при $n = 2$ имеем операцию декартового произведения двух множеств $M \times M'$, как совокупность всех упорядоченных пар (a, a') , где $a \in M, a' \in M'$. Определим N -арное отношение как произвольное подмножество декартового произведения $M_1 \times M_2 \times \dots \times M_n$. Отсюда следует, что бинарное отношение есть любая совокупность упорядоченных пар вида (a, a') .

Определение клона функциональных n -отношений связано с операцией суперпозиции функций, ассоциированных с указанными отношениями. Пусть F и G — бинарные отношения, с которыми связаны одноместные функции f и g . Суперпозиция (композиция) отношений $F \times G$ порождает новое отношение H , с которым связана одноместная функция $h(x) = f(g(x))$, являющаяся указанной суперпозицией функций f и g . Иными словами, пара $(a, s) \in H$, если существует вспомогательный элемент b такой, что пара $(a, b) \in F$, а пара $(b, s) \in G$. Таким образом, b — значение функции g на элементе s , тогда как a — значение функции f на результате применения функции g при значении ее аргумента s .

Приведенные определения суперпозиции бинарных отношений и одноместных функций можно обобщить на случай n -отношений и $(n-1)$ -арных функций. Определим обобщенную свертку де Моргана (прямой угол) — суперпозицию n -арных отношений и $(n-1)$ -арных функций. Пусть $F, G_1, G_2, \dots, G_{n-1}$ — n -арные функциональные отношения, с каждым из них связана $(n-1)$ -местная функция f, g_1, \dots, g_{n-1} . К суперпозиции $H = F(G_1, G_2, \dots, G_{n-1})$ принадлежит n -ка (a_1, a_2, \dots, a_n) , если найдутся такие вспомогательные элементы b_1, b_2, \dots, b_{n-1} , что

$$\begin{array}{cccccc} (a_1 & a_2 & \dots & a_{n-1} & b_1) & \in & G_1 \\ (a_1 & a_2 & \dots & a_{n-1} & b_2) & \in & G_2 \\ \dots & \dots & \dots & \dots & \dots & & \dots \\ (a_1 & a_2 & \dots & a_{n-1} & b_{n-1}) & \in & G_{n-1} \\ (b_1 & b_2 & \dots & b_{n-1} & a_n) & \in & F \end{array}$$

Определим i -суперпозицию (крест) как обобщение свертки де Моргана. Пусть $G_1, G_2, \dots, G_{i-1}, G_i, G_{i+1}, \dots, G_n$ — функциональные отношения, с каждым из них связана $(n-1)$ -местная функция $g_i(x_{i1}, x_{i2}, \dots, x_{in-1})$, где $i = 1, 2, \dots, n$.

Как обобщение композиции бинарных отношений введем S_i -суперпозицию, так порождающую новое n -отношение $W = S_i(G_1, G_2, \dots, G_n)$, что n -ка $(a_1, a_2, \dots, a_n) \in W$, если существуют $n-1$ вспомогательных элементов: $b_1, b_2, b_{i-1}, b_{i+1}, \dots, b_n$, для которых выполняются принадлежности:

$$\begin{array}{cccccccc} (a_1 & a_2 & \dots & a_{i-1} & b_1 & a_{i+1} & \dots & a_n) & \in & G_1 \\ (a_1 & a_2 & \dots & a_{i-1} & b_2 & a_{i+1} & \dots & a_n) & \in & G_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \dots \\ (a_1 & a_2 & \dots & a_{i-1} & b_{i-1} & a_{i+1} & \dots & a_n) & \in & G_{i-1} \\ (b_1 & b_2 & \dots & b_{i-1} & a_i & b_{i+1} & \dots & b_n) & \in & G_i \\ (a_1 & a_2 & \dots & a_{i-1} & b_{i+1} & a_{i+1} & \dots & a_n) & \in & G_{i+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \dots \\ (a_1 & a_2 & \dots & a_{i-1} & b_n & a_{i+1} & \dots & a_n) & \in & G_n \end{array}$$

При $i = n$ S_i -суперпозиция образует свертку де Моргана.

Теорема 1. Операция i -суперпозиции $S_i(G_1, G_2, \dots, G_n) = W$ функциональных n -отношений порождает новое функциональное отношение W , сопряженное с ассоциированной суперпозицией функций.

В [6] рассмотрены следующие операции над n -отношениями: проекция, фильтрация, соединение, деление, теоретико-множественное пересечение, объединение, разность, декартово произведение, которые широко используются при построении реляционных баз данных. В зависимости от вида представительной алгебры n -отношений получаем различные клоны n -отношений. Такие клоны формализуют процесс построения как структур данных, так и алгоритмов их обработки, реализуемых операторной составной клона. При этом логическая компонента клона является связующим звеном в клоне, поскольку она используется как для построения баз данных, так и для описания логической структуры алгоритма. Назовем клоном Кодда клон, в котором в качестве основы представительной алгебры используется реляционная алгебра.

При введении в клон Кодда операций альтернативы и прогнозирования получаем более широкие возможности для построения структур данных и соответственно более мощный клон n -отношений. Интересным представляется введение как в логическую компоненту представительной алгебры, так и в алгебру n -отношений кванторов существования и всеобщности. В результате расширяются возможности не только построения баз данных и обрабатываемых их приложений, но и исследований эквивалентных преобразований с целью оптимизации таких построений.

Исследование проблемы функциональной полноты клона n -отношений позволило установить факт изоморфизма такого клона клону регулярных событий Клини. Как следствие, это позволило использовать технику построения решетки подалгебр в клоне Клини для клона n -отношений.

Теорема 2. Клон n -отношений изоморфен клону Клини.

Следствие. Теория клонов n -отношений служит математическим фундаментом в прикладном программировании при построении реляционных баз данных и знаний.

Построение клона n -отношений и введение в сигнатуру операций представительной алгебры кванторов существования и всеобщности позволяет решить проблему построения нормальных канонических форм n -отношений. Преобразование выражений n -отношений к канонической форме решает проблему эквивалентности в таком клоне и позволяет стандартизовать вид запросов в базах данных, их оптимизацию и преобразование.

В клоне n -отношения актуален вопрос взаимосвязи логической, алгоритмической и табличной компонент. При этом предлагается использовать разметку. По табличной компоненте клона n -отношения передвигаются указатели, каждый из которых связан с циклически применяемым по месту указателя оператором обработки строк или столбцов таблицы, в то же время сама алгоритмическая схема передвижения указателей задается САА-схемой более высокого уровня.

КЛОН n -ОТНОШЕНИЙ ДЛЯ ЗАДАЧ ОПЕРАТИВНОГО НАКОПЛЕНИЯ И ФИНАНСОВОГО АНАЛИЗА

В качестве иллюстрации формализованного проектирования с использованием клона n -отношений рассмотрим задачу первичного структурированного накопления информации в экономических задачах как результат выполнения хозяйственных или другого рода операций. Предложенную далее модель назовем учетной. Предположим, что прикладной пользователь имеет возможность легко модифицировать эту модель с целью настройки ее для решения задач учета конкретной предметной области (бухгалтерского, финансового, управленческого, статистического и других видов учета для объектов различной природы). Основными элементами такой модели являются счета, аналитические объекты, журнал регистрации операций, документы для регистрации операций, отчеты.

В счетах как учетных регистрах накоплены итоговые данные по операциям, которые формируются с учетом временного интервала проведения операции. Такие итоговые данные могут быть разделены на составляющие в соответствии с объек-

тами, участвующими в операции. Например, счет «товар» может содержать все количество имеющихся на складе товаров или разбиваться на составляющие по каждому товару. Тогда говорят о наличии аналитического учета по конкретному счету, а объекты разбиения итоговых данных называют аналитическими. В предложенной модели каждую операцию регистрируют соответствующими ей документами в журнале операций. Документы реализуют регистрацию входной информации учетной операции, алгоритм формирования итоговых данных в счетах, печатную форму регистрации такой операции.

Создание документов осуществляется либо самим пользователем, либо перенасройкой готовых поставляемых системой документов. Пользователь самостоятельно может определить структуру входной информации, характеризующую операцию. Для экономических задач входная информация задается обычно табличной ведомостью, состоящей из шапки и таблицы. Каждая строка таблицы обрабатывается однотипно, имеет единую структуру и связана коддовой операцией соединения с полями шапки ведомости. Для каждой строки таблицы можно определить арифметические формулы расчетов полей (аналогично электронным таблицам), описать алгоритмическую схему формирования итогов по счетам и создать печатную форму документа регистрируемой операции. Весь процесс создания документов поддерживается инструментарием.

Отчеты отображают итоговые данные по разным временным периодам, разбиение итоговых данных по аналитическим объектам, операциям, взаимодействию между отдельными аналитическими объектами. Отчеты используются не только для анализа итоговых данных, но и для редактирования процесса регистрации операций. Основу модели составляют n -отношения:

R_1 — план счетов; данное отношение определено на декартовом произведении следующих множеств: <номера счетов>*<наименования счетов>*<группы аналитических объектов, связанных с данным счетом>; множество <группы аналитических объектов> может быть пустым, в этом случае для выбранного счета не ведется аналитический учет;

R_2 — группы аналитических объектов (назовем их справочниками); это отношение определено на декартовом произведении множеств <номера справочников>*<наименования справочников>*<атрибуты (свойства) аналитического объекта>; в данном случае <атрибуты аналитического объекта> могут быть множеством или отношением, если количество атрибутов данного объекта больше единицы (например, аналитический объект «организация» характеризуется набором атрибутов, которые могут быть описаны в виде следующего отношения: <наименование организации>*<юридический адрес>*<банковский расчетный счет> и т.д.);

R_3 — журнал регистрации учетных операций; это отношение определено на следующем декартовом произведении: <дата операции>*<наименование документа, регистрирующего операцию>*<состояние операции — осуществленная или планируемая>;

R_4 — документы регистрации операций; данное отношение определено следующим образом: <номер документа>*<структура входной информации для документа>*<программа формирования накопления итогов на счетах, затрагиваемых данной операцией>*<печатная форма отчета по данной операции>. Здесь <структура входной информации для документа> представляет собой отношение, описываемое в виде таблицы структуру входной информации, которая должна быть заполнена и рассчитана в процессе регистрации операции. Форма отчета по операции также определяется в виде отношений, тогда как программа формирования итоговых результатов по факту регистрации данной операции может быть описана САА-схемой алгоритма ее функционирования над входными данными документа и итоговыми данными учетной модели. Перечисленные отношения определяют структуру данных функционирования такой модели, а алгоритмы регистрации и накопления информации описываются в виде САА-схем, по которым с помощью инструментальных средств строятся программы в выбранном языке программирования. Инструментальные средства алгоритмики, кроме автоматизированных средств построения программ по их описаниям в САА-схемах, содержат средства

преобразования n -отношений описания данных в предметной области в структуру той базы данных, в среде которой будет осуществляться проектирование программной системы функционирования такой модели. Манипулирование данными, представленными в виде n -отношений, сводится к использованию перечисленных в разд. 1 операций в клоне n -отношений и автоматическому синтезу по выражениям в такой алгебре соответствующих операторов языка SQL.

Создание системы проектирования учетной модели непрофессиональным программистом (предметным специалистом) соответственно его потребностям и наличие средств синтеза таких приложений вызвано разнообразием учетных моделей и динамикой их изменения. В настоящее время можно констатировать наличие на рынке автоматизации экономических задач в основном универсальных систем, пытающихся свести разнообразие учетных моделей к общей схеме, что требует больших ресурсов. При этом роль предметного пользователя сводится к попытке постановки задачи адаптации без соответствующих знаний о самой адаптируемой системе. В то же время создание для каждого предприятия специализированной системы учета требует не меньших ресурсов, чем адаптация. Рассмотренный в статье подход предполагает создание постоянно дополняемой базы алгоритмических знаний для генерации и функционирования учетных моделей. Перечислим компоненты, из которых состоит база знаний.

- **Информационная компонента:** объекты учетной модели представлены с использованием алгебры n -отношений в табличной форме в среде визуальных инструментов. Обобщенную информационную учетную структуру предметный пользователь настраивает на учетную область.

- **Функциональная компонента:** состоит из схем алгоритмов реализации расчетов в предметной области. Предоставляются накопленные библиотеки стандартных бухгалтерских, финансовых, статистических расчетов в виде САА-схем, которые представлены в хорошо прокомментированной естественно-лингвистической или граф-схемной форме. Узлы граф-схем, кроме алгоритмической нагрузки, несут семантическую нагрузку в виде правовой или управленческой информации, обновляемой администратором информационной компоненты.

- **Структурированные библиотеки реализаций** интерпретаций схем формализуются на языках программирования. Конечные узлы граф-схем — достаточно независимые программные компоненты на целевом языке реализации учетной модели. Такие реализации создают программисты, интерпретируя элементарные логические условия и операторы, типичные средства отображения и манипулирования табличными данными (элементарные n -отношения); в комплексе с самой САА-схемой алгоритма обработки строится приложение.

- **Специализированные средства создания и модификации алгоритмов расчета** (документов) в учетных моделях: кроме алгоритмической структуры, пользователь задает информационную структуру задачи из находящихся в ней данных и формируемых промежуточных результатов в виде n -отношения (табличная форма) с привязкой алгоритмов расчета. Разработанные документы пополняют функциональную компоненту. Содержание документов учетных моделей:

- структура входной регистрируемой информации (в виде таблицы с перечнем наименований и типов полей таблицы и шапки экономической ведомости);
- алгоритмы расчета значений в обрабатываемой ведомости (в вырожденном случае формула расчета подобна Excel-формулам, в более сложных случаях задается САА-схема);
- алгоритм изменения итоговых данных учетной модели на основании новой регистрационной информации, задаваемой документом;
- печатная форма учетной операции, которая документально фиксирует ее проведение.

- **Библиотека стандартных отчетов по итогам накопительных счетов** с выходом в режим редактирования при регистрации учетной операции. Библиотека легко адаптируется к структуре накопительных счетов и привязке к ним аналитических объектов, называемых планом счетов. Такой план задает структуру учетной модели в привычной для экономистов табличной форме и легко корректируется непрофессиональным пользователем.

Определяя структуру счетов, перечень аналитических объектов и их свойства, характеристики журнала операций и стандартных отчетов, модифицируя и создавая новые документы обработки, предметный пользователь строит свою учетную модель на клоне n -отношений.

Важным инструментом дальнейшего анализа накопленных оперативных данных является компонента анализа. В нее входят алгоритмы статистического анализа и анализа с использованием нейронных сетей в случае неоднородности оперативных данных. С этой целью формируется база алгоритмических знаний о статистических методах обработки данных и методах нейронных сетей.

ИНСТРУМЕНТАРИЙ ПОДДЕРЖКИ ТЕОРИИ КЛОНОВ

Представленный в настоящей статье математический аппарат позволяет исследовать алгоритмическую структуру знаний предметных областей и определять важные для предметной области алгоритмические зависимости, порождать новые алгоритмические знания, устанавливать взаимосвязи между близкими предметными областями. В то же время он положен в основу инструментария создания программных систем. Аппарат алгебры алгоритмики предлагает высокоуровневые средства проектирования алгоритмов и структур данных для специалистов предметных областей, не владеющих в совершенстве техникой объектно-ориентированного программирования. Эти средства составляют основу восходящего от синтезатора Мультипроцессист [7, 8] интегрированного инструментария проектирования и синтеза алгоритмов и программ для погружения в объектно-ориентированную среду. Дальнейшее развитие алгебры алгоритмики для построения клонов n -отношений и алгебр, ориентированных на формализацию и проектирование структур данных для предметных областей, служит мощным стимулом ее развития.

В [9] описан интегрированный инструментарий проектирования и синтеза (ИПС) алгоритмов и программ. Синтезатор Мультипроцессист — одна из первых украинских CASE-систем, ориентированная на генерацию программ. В ее основу положен метод многоуровневого структурного проектирования программ (МСПП) по их описаниям в языке САА/1. В отличие от Мультипроцессиста, ориентированного на синтаксический анализ САА-схемы, в ИПС схемы проектируются на основе метода диалогового конструирования синтаксически правильных программ (ДСП-метода), обеспечивающего синтаксическую правильность схемы. Особенность ИПС — интеграция трех форм представления алгоритмов при их проектировании: аналитического, естественно-лингвистического и графового.

В состав ИПС входят следующие компоненты (рис. 1):

— ДСП-конструктор для диалогового проектирования синтаксически правильных схем последовательных и параллельных алгоритмов и генерации программ, в том числе программ для объектно-ориентированных сред (в частности, C++ и Java);

— ДСП-конструктор граф-схем алгоритмов;

— ДСП-конструктор табличных структур данных (n -отношений);

— редакторы САА-схем, граф-схем, таблиц;

— трансформатор для интерактивной трансформации схем алгоритмов и программ с целью их улучшения по различным критериям (используемая память, время выполнения);

— генератор САА-схем по схемам более высокого уровня — гиперсхемам [17], представляющим обобщение грамматик структурного проектирования;

— среды конструирования алгебро-алгоритмических описаний (СКАО), которые содержат

- описание конструкций САА, операций алгебры n -отношений в терминах целевого языка проектируемого приложения;
- базисные понятия и их программные реализации;
- метаправила конструирования алгоритмов (свертка, развертка, трансформация, переориентация);
- схемы алгоритмов (базисные понятия заменены абстрактными переменными), стратегии обработки (фиксируют алгоритмическую структуру обработки) и гиперсхемы.



Рис. 1. Архитектура инструментария ИПС

Основное предназначение ДСП-конструктора — поуровневое конструирование схем сверху вниз с детализацией языковых конструкций САА. Проектирование отражает дерево конструирования алгоритма. На каждом шаге конструирования система в диалоге с пользователем предоставляет на выбор лишь те конструкции, подстановка которых в формируемый текст не нарушает синтаксической правильности схемы. По полученному в результате конструирования дереву алгоритма можно синтезировать программы на основе программных реализаций из СКАО.

Конструктор табличных структур данных ДСП помогает визуальными средствами формировать таблицы баз данных и на их основе конструировать новые, как результат выполнения операций клона n -отношений над ними.

ДСП-конструктор граф-схем строит граф-схему алгоритма из базовых конструкций операций клона Калужнина на тех же принципах, что ДСП-конструктор САА-схем.

Алгебраические средства проектирования алгоритмов и структур данных соответствуют принятому в объектно-ориентированных средах конструированию объектов в терминах шаблонов. При этом поуровневое присвоение логическим и операторным переменным их интерпретаций суть заполнение полей шаблонов в рамках объектных сред. Отметим использование алгебраических преобразований схем на основе алгебраических операций, входящих в формулы, представляющие проектируемые объекты.

Наряду с алгебраической формой представления объектов сочетание их адекватных текстовых и графовых описаний способствует анализу взаимодополняющих аспектов проектируемых объектов и соответствующих средств автоматизированного синтеза. Так, граф-схемная форма проектирования адекватна визуальному представлению знаний в UML с использованием объектно-ориентированного программирования в Rational Rose.

Для поддержки в базе знаний проектов из разных предметных областей существует возможность установления между всеми ее компонентами иерархических связей разной природы: подчиненности, взаимозаменяемости, наследования и т.п. Для формализации таких иерархий используются LL(1)-грамматики, вывод в которых осуществляется так называемой САА-машиной. Дополнительно САА-машина выполняет синтаксический анализ САА-схем, созданных без использования

ДСП-конструктора, синтезирует и собирает программы по их САА-схемам и интерпретациям базовых понятий. Такая САА-машина представляет универсальный механизм работы с иерархическими структурами.

САА-машина реализована как абстрактный автомат, выполняющий набор команд. Программой для нее служит LL(1)-грамматика, описывающая синтаксис языка, на принадлежность к которому анализируется входная символьная строка. Ввиду близости форм представления формул САА-схем и LL(1)-грамматик автомат способен интерпретировать САА-схемы. Поэтому описание команд автомата рассмотрим с двух точек зрения: реализации алгоритма синтаксического анализа по LL(1)-грамматикам и интерпретации САА-схем. Для сравнения приведем примеры фрагментов программ автомата LL(1)-грамматик, внутреннего представления формулы САА-схемы и соответствующей формулы САА-схемы. Здесь команды L, K проверяют и считывают терминалы из входной строки для синтаксического анализа; A1, A2 — семантические процедуры трансляции; B1, B2 — нетерминалы; X — команда, завершающая альтернативную ветвь распознавания; XL — завершение логического выражения. Схемы во втором и третьем столбцах таблицы эквивалентны. Логические операции конъюнкции и дизъюнкции в первом столбце представлены символами &, !. Команда C в САА-машины реализует циклическую обработку. Важным моментом работы САА-программ, как и САА-схем, является наличие системы перемещающихся указателей и маркеров, положение которых в структуре данных задает место работы элементарного оператора или условия, т.е. связь по данным между процедурами обработки в САА-схемах осуществляется как через механизм передачи параметров, так и по месту расположения указателей и маркеров. Это позволяет естественно представлять рекурсивные алгоритмы посредством итеративных операций [10].

Программы автомата интерпретации		Соответствующая формула САА-схемы
LL(1)-грамматика	САА-схемы	
B1=L'схема' B2 A1 X ; K'объект' B1 A2 X ; E X ; B2=&L'конец'K'='XL A2 X ; E X ;	B1=!V1 V2 XL A1 X ; E B2 X ; B2=C V3 XL A2 X ;	B1= ЕСЛИ V1+V2 ТО A1 ИНАЧЕ B2 ; B2= ПОКА НЕ V3 ЦИКЛ A2

САА-схема преобразуется в списковую внутреннюю форму и становится входной символьной строкой, проход по которой вызывает интерпретацию САА-машины. Механизм САА-машины в новой версии инструментария усовершенствован и используется для обхода любых иерархических структур.

СИСТЕМА ОРГАНИЗАЦИИ ДИСТАНЦИОННОГО И СТАЦИОНАРНОГО ОБУЧЕНИЯ «ВИРТУАЛЬНАЯ АУДИТОРИЯ»

Рассмотрим пример использования клона *n*-отношений для создания web-приложений в системе «Виртуальная аудитория» для организации учебного процесса в стационарной и дистанционной формах. Система снабжена средствами озвучивания, и поэтому ее целесообразно использовать людям с проблемами зрения. В основу методики обучения положен метод СПРИНТ интенсивного взаимодействия преподавателя с аудиторией в режиме обратной связи. Предполагается наличие соответствующего методического (МО) и презентационного (ПО) обеспечения.

В МО входят подготовленные преподавателем конспекты лекций по всем преподаваемым дисциплинам. Конспект каждой лекции организован как многоуровневая гипертекстовая структура с подключением словаря базовых понятий дисциплины. Древовидный текст лекции предполагает ее нелинейное изучение, навигацию

по тексту лекции, исходя из подробного плана лекции, что особенно важно для людей с проблемами зрения и для дистанционного обучения, поскольку организованная таким образом лекция отображает методологическую подчиненность изучаемого материала. Согласно СПРИНТ-технологии конспект каждой лекции предоставляется учащимся для предварительного самостоятельного изучения как в изданном виде, так и в электронном, затем на лекции закрепляется изученный материал путем диалога с преподавателем.

Во время лекции независимо от стационарной или дистанционной формы обучения кадры последовательно высвечиваются на экран (общий чат в виртуальной аудитории). Кадры, как отдельные информационные фрагменты, предназначены для обучения и включают понятийный аппарат, конструирование составных понятийных объектов из составляющих, а также примеры, упражнения и т.д.

При выводе на экран очередного кадра лекции обучаемый может задавать преподавателю вопросы, возникшие у него в процессе предварительного ознакомления с данным кадром. После ответов на поставленные вопросы преподаватель персонально адресует свои вопросы учащимся. Активность обучаемых (их успехи и неудачи) фиксируется для объективной оценки знаний обучаемых в процессе лекции, а затем и интегральной их оценки на разных этапах освоения курса в целом.

В разработке проекта для дистанционных и стационарных форм преподавания используется ИПС. Проект «Виртуальной аудитории» автоматизирует учебный процесс произвольного учебного заведения (и для студентов с проблемами зрения). Параллельно с созданием накоплена база знаний для web-приложения и выделены такие структурные элементы:

- виртуальная аудитория;
- электронная библиотека;
- система формирования учебного плана для каждого студента (обязательные дисциплины и дисциплины по выбору);
- автоматизированный учет успеваемости студентов;
- система формирования учебного расписания;
- администрирование системы, поддержка прав учебно-организационных подразделений вуза, преподавателей и студентов.

Администрирование «Виртуальной аудитории» включает:

- регистрацию студентов на основании приказа на их зачисление;
- регистрацию преподавателей;
- регистрацию педагогической нагрузки в каждом учебном семестре на основе учебных планов кафедр;
- регистрацию административного и обслуживающего персонала (работников деканатов, кафедр, учебной части и т.п.).
- поддержку работоспособности системы.

Виртуальная аудитория поддерживает дистанционный процесс обучения как чат с возможностью организации дополнительных комнат для индивидуальной работы со студентами по выбору преподавателей. Для виртуальной аудитории составляется расписание занятий с указанием темы, времени проведения и фамилий преподавателей. Для студентов открыт доступ для записи на обучение. На занятии одновременно может присутствовать не более восьми студентов.

Электронная библиотека включает:

- гипертекстовый редактор для подготовки учебных материалов;
- структуру библиотеки, отражающей учебный процесс — список выпускаемых специальностей, обязательных учебных дисциплин и дисциплин за выбором.

Электронная библиотека имеет иерархическую структуру вида:

- 1-й уровень — перечень специальностей;
- 2-й уровень — перечень учебных дисциплин;

— 3-й уровень — учебный план по каждой дисциплине: темы лекционных и лабораторных занятий, формы контроля и контрольные тесты, материалы для самостоятельного обучения;

— 4-й уровень — учебно-методический материал по каждой теме.

Материал лекций структурирован следующим образом.

План лекции. Основной лекционный материал организован как набор текстовых кадров, вызываемых на экран при активизации студентом определенного пункта плана, связанного с таким кадром. В кадре цветом выделены понятия или тезисы, которые появляется во всплывающем окне при подводе курсора. Такая иерархическая гипертекстовая структура лекции упрощает навигацию по конспекту лекций, что имеет важное значение для незрячих студентов.

В учебные материалы входят: тезаурус, который представляет перечень основных понятий темы, оформленный в виде гиперссылок из текста лекций; аббревиатура — составная часть этого тезауруса; список рекомендуемой литературы; перечень типичных вопросов и ответов на них (накопительная база); практические методические указания; решения типичных задач по теме; перечень практических заданий для самостоятельного решения; контрольные тесты, которые выполняются в режиме online; контрольные работы.

Виртуальная аудитория, представленная в данной статье, аккумулирует опыт построения алгоритмических баз знаний для автоматизированного создания произвольных web-приложений. В то же время она актуальна сама по себе как программный продукт для дистанционного обучения студентов, в том числе с проблемами зрения.

СПИСОК ЛИТЕРАТУРЫ

1. Многоуровневое структурное проектирование программ: Теоретические основы, инструментарий / Е.Л. Ющенко, Г.Е. Цейтлин, В.П. Грицай, Т.К. Терзян. — М.: Финансы и статистика, 1989. — 208 с.
2. Цейтлин Г.Е. Введение в алгоритмику. — К.: Сфера, 1998. — 310 с.
3. Czarnеcki K., Eisenecker U. Generative programming: methods, tools, and applications. 1st edn. Addison-Wesley Professional, 2000. — 864 p.
4. Кон П. Универсальная алгебра. — М.: Мир, 1968. — 348 с.
5. Курош А.Г. Лекции по общей алгебре. — М.: Физматгиз, 1962. — 396 с.
6. Сodd E.F. A relational model of data for large shared data banks // Comm. ACM. — 1970. — 13, № 6. — P. 377–387.
7. Грицай В.П., Захарія Л.М. Мультипроцесист — автоматизована система багаторівневого проектування програм // Тези доп. 9-ї Всесоюз. наради з питань управління. — Єреван, 1983.
8. Захарія В.П., Захарія Л.М. Інструментарій синтезу програм для міні- і мікро ЕОМ // Кибернетика. — № 6. — 1989.
9. Яценко Е.А., Мохница А.С. Инструментальные средства конструирования синтаксически правильных параллельных алгоритмов и программ // Пробл. программирования. — 2004. — № 2–3. — С. 444–450.
10. Захарія Л.М. Некоторые математические и программные средства создания специализированных экспертных систем // Кибернетика и системный анализ. — 1997. — № 4. — С. 182–186.
11. Цейтлин Г.Е. Проблема функциональной полноты для метаалгебр регулярных событий // Кибернетика и системный анализ. — 2000. — № 6. — С. 14–27.

Поступила 08.07.2009