

Никифоров Андрей Юрьевич

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СИНТЕЗА ЯЗЫКОВ И ПРОТОКОЛОВ
ВЗАИМОДЕЙСТВИЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Специальность 05.13.11 – математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Автор:



Работа выполнена в Московском инженерно-физическом институте
(государственном университете)

Научный руководитель: кандидат технических наук, доцент
Русаков Виктор Анатольевич

Официальные оппоненты: доктор технических наук, профессор
Абросимов Леонид Иванович

кандидат технических наук, доцент
Федоров Николай Владимирович

Ведущая организация: Московский государственный горный
университет

Защита состоится 1 июля 2009 года в 15 час. 00 мин. на заседании диссертационного
совета Д212.130.03 в Московском инженерно-физическом институте (государственном
университете) по адресу: Москва, Каширское ш., 31.

С диссертацией можно ознакомиться в библиотеке Московского инженерно-
физического института (государственного университета).

Отзывы в двух экземплярах, заверенные печатью, просьба направлять по адресу:
115409, г. Москва, Каширское ш., 31, Диссертационные советы МИФИ.

Автореферат разослан « » мая 2009 года.

Ученый секретарь
диссертационного совета



Шумилов Ю.Ю.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Для подготовки квалифицированных кадров и обеспечения долгосрочных потребностей инновационной экономики в настоящее время создается сеть федеральных университетов в целях системной модернизации высшего образования и комплексного развития регионов на основе интеграции науки, образования и производства. Для модернизации образовательной деятельности требуется разработка образовательных контентов и распределенных ресурсов нового поколения, внедрение новых видов и технологий обучения, усиление роли дистанционного обучения.

В пропаганду, научные исследования и внедрения идей дистанционного образования внесли вклад А.А. Андреев, В.П. Тихомиров, А.Д. Ивашников, Ю.А. Чернышев, В.В. Семенов, А.Г. Шмелев, В.П. Невежин, А.А. Золотарев, И.П. Рождественский и др.

Распределенные системы и коммуникационные протоколы являются одной из наиболее быстро развивающихся областей современного программирования. Имеющиеся средства упорядоченного описания взаимодействия систем разнородны, зачастую ориентированы на конкретные приложения и неоднозначны в интерпретациях. Такие описания объемны, трудоемки в изучении и слабо адаптируются к требованиям участников взаимодействия.

Так, например, основные положения семиуровневой эталонной модели взаимодействия открытых систем включают в себя 5 книг на 120 страницах и формируют рамки применения более чем 150 стандартов.

Традиционными средствами описания являются универсальные и специализированные языки. Первые – сложны в изучении и неудобны в применении, несущественные языковые подробности затушевывают важные аспекты описания взаимодействий. Вторые – громоздки и имеют узкую область применения.

Например, изучение языка C/C++ занимает до 200 часов, а описания языков SDL-2000 и E-LOTOS занимают около 250 и 200 страниц соответственно.

В публикациях Malloy В., Power J. и др. был предложен подход к измерению сложности языков на основе метризации грамматик. Его использование позволяет заложить базис для выделения класса языков низкой описательной сложности (LDC – Low Descriptive Complexity).

Создание методов и инструментальных средств синтеза LDC языков описания взаимодействия распределенных систем позволяет решить данную проблему, что указывает на актуальность темы разработки методов и инструментальных средств синтеза языков и протоколов взаимодействия распределенных систем.

Цель исследований. Целью диссертации является создание инструментальных средств синтеза языков и протоколов взаимодействия распределенных систем.

Для достижения поставленной цели в диссертации решены следующие основные задачи:

1. Исследованы современные модели и методы управления индивидуализацией сред поддержки обучения.
2. Проанализированы существующие подходы к оценке сложности языков. В результате анализа предложен класс LDC языков.
3. Разработаны модели обучения и студента.
4. Проведена разработка архитектуры индивидуализированных языковых инструментальных сред поддержки обучения.
5. Разработан метод создания LDC языков.
6. Создан метод генерации языковых инструментальных сред.
7. Синтезирован LDC язык упорядоченного описания взаимодействия систем.
8. Разработанный язык адаптирован к применению на примере обучения по курсу «Взаимосвязь открытых систем» (ВОС), а также разработано средство его индивидуализации.
9. Реализована среда поддержки лабораторных занятий по курсу ВОС.
10. Апробирована среда поддержки лабораторных занятий по курсу ВОС.

Объект исследования. Объектом исследования является синтаксис процедурных языков программирования.

Предмет исследования. Предметом исследования являются метрики грамматик, языковые инструментальные среды.

Методы исследования. Для решения поставленных задач в диссертации использованы теория графов, теория множеств, теория формальных языков и теория построения трансляторов. При разработке программного обеспечения использованы методы объектно-ориентированного программирования.

Научная новизна результатов работы. В диссертации получены следующие новые результаты:

1. Разработан метод создания языков LCD класса, основанный на метризации грамматик.
2. Синтезирован базовый язык упорядоченного описания взаимодействия систем и доказана его алгоритмическая полнота.
3. Предложен оригинальный метод генерации инструментальных сред с порождением семейства пользовательских языков.
4. Разработаны инструментальные программные средства генерации сред моделирования упорядоченного взаимодействия, с использованием которых построены прототипы протоколов и элементов пользователя различных распределенных систем.

Практическая значимость результатов работы. Использование языков низкой описательной сложности в программном обеспечении позволяет упростить его освоение, реализацию и поддержку. Использование индивидуализированных сред позволяет ускорить их освоение и

результативность работы, в обучении снизить нагрузку на преподавателей и повысить самостоятельность выполнения работ студентами.

На защиту выносятся:

1. Разработанный класс LDC языков упорядоченного описания взаимодействия систем.
2. Метод генерации множества LDC языковых инструментальных сред.
3. Архитектура индивидуализированных языковых инструментальных сред поддержки обучения.
4. Программная реализация инструментальных средств генерации множества языковых инструментальных сред.
5. Инструментальные средства оценки протоколов.

Достоверность полученных результатов. Достоверность научных результатов подтверждается корректным применением выбранных методов исследования и экспериментальными данными тестирования разработанных программных средств, соответствующими теоретическим выводам.

Апробация работы. Основные положения и результаты диссертации докладывались и обсуждались на следующих конференциях и семинарах: XVI Международный научно-технический семинар «Современные технологии в задачах управления, автоматизации и обработки информации» (Алушта, 2007); XII Московская международная телекоммуникационная конференция студентов и молодых ученых "Молодежь и наука" (Москва, 2008); XV международная научно-техническая конференция студентов и аспирантов "РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА" (Москва, 2009); «Научная сессия МИФИ» (Москва, 2001 – 2005, 2007).

Реализация результатов работы. Результаты, полученные в работе, имели следующие научно-практические приложения:

- Созданный LDC язык описания и реализации протоколов и инструментальные средства разработки и тестирования протоколов использованы в проекте "ИнфраМенеджер" ЗАО «СофтИнтегро», что подтверждается актом об использовании результатов диссертации.
- Разработанный программный комплекс поддержки лабораторных занятий по курсу «Взаимосвязь открытых систем» внедрен в учебный процесс МИФИ, что подтверждается актом об использовании результатов диссертации.

Публикации. Основные результаты диссертации опубликованы в 11 печатных работах, в том числе статья в журнале, включенном ВАК РФ в перечень ведущих рецензируемых научных журналов и изданий.

Структура работы. Диссертация состоит из введения, пяти глав, заключения, списка использованной литературы из 107 наименований и приложений. Основная часть диссертации содержит 130 страниц машинописного текста, включая 34 рисунков, 13 таблиц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, её научная новизна и практическая значимость, сформулирована цель работы.

В первой главе проводится анализ подходов к созданию компьютерных систем обучения и методов индивидуализации средств поддержки, анализ языков формального описания и способы оценки описательной сложности языков.

Развивающаяся в настоящее время технология компьютерного обучения подчиняется общим закономерностям процесса. В 1999 году Лорин Андерсон и его коллеги опубликовали обновленную версию Таксономии Блума. В ней разделяются измерение *Знания* и измерение *Когнитивных Процессов*. В измерении *Когнитивных Процессов* выделяются следующие уровни: знание, понимание, применение, анализ, оценка и создание. В измерении *Знания* выделяются фактическое знание, концептуальное знание, процедурное знание и метакогнитивное знание. В соответствии с этой таксономией каждый уровень знания может соотноситься с каждым уровнем когнитивного процесса. Лишь самостоятельная работа превращается в научно-исследовательскую, охватывая все этапы процесса обучения-познания.

Можно выделить следующие основные подходы к индивидуализации образования:

- генеративный – необходимые учебные задания создаются исходя из актуальных задач на основе имеющихся баз данных и знаний;
- генеративно-диагностический – создаются актуальные задания по результатам предварительной диагностики характеристик студента;
- поисковый (навигационный) – содержание образования предлагается поисковыми системами на основе запроса пользователя;
- эвристический – в самом общем виде эвристики представляют собой группы специально организованных вопросов, отвечая на которые студент развивает свои способности;
- экспертный – по мере продвижения студента по ступеням саморазвития программные средства предлагают ему конкретные рекомендации по управлению саморазвитием, направленные на обеспечение решения образовательных задач.

Модель студента должна удовлетворять следующим требованиям:

- релевантность – система должна учитывать именно те индивидуальные особенности студента, которые существенны для достижения намеченных образовательных целей;

- адекватность – система должна обеспечить соответствие модели студента её оригиналу, здесь важно разделение устойчивых и ситуативных индивидуальных особенностей;
- состоятельность – уточнение модели за счет накопления данных о студенте.

Важность формального описания была осознана на ранних этапах развития сложных, слабо формализованных систем. Первоначально описания носили неформальный, словесный характер. Однако практика очень быстро убедила разработчиков в обратном – субъективная природа восприятия словесных описаний не позволяет согласовать разрабатываемые стандарты, описания носят неоднозначный характер, не обладают полнотой, не имеют формальной основы для анализа. И, что самое главное, приводят к несовместимости дорогостоящих изделий и не могут служить долгосрочной основой для развития сложных распределенных систем. Выход из создавшегося положения может быть найден только при использовании методов формального описания. Одним из наиболее популярных методов описания стали алгоритмические языки.

Анализ широко известных и применяемых языков, таких как Estelle, LOTOS, SDL, позволил выделить следующие основные требования к языку упорядоченного описания взаимодействия систем. Такой язык должен быть, во-первых, пригоден для формального описания статичности и динамики компонентов архитектуры упорядоченного взаимодействия систем. Во-вторых, он должен быть инвариантен относительно существующих архитектур взаимосвязи.

Для оценки сложности языков, влияющей на стоимость обучения и поддержки использующего их программного обеспечения, в работах различных авторов введены метрики. Потенциально эти метрики можно использовать при создании новых языков, обеспечивая их простоту.

Сформулирована цель и поставлены конкретные задачи диссертационного исследования.

Во второй главе описываются разработанные модели обучения и студента, а также архитектура индивидуализированных языковых инструментальных сред поддержки обучения.

Входом процесса обучения являются содержание и модели требуемых знаний предмета, а выходом – уровни знания, формируемые в сознании студента. Содержание и модели требуемых знаний предмета формирует преподаватель.

Управляющим звеном в управлении процессом обучения является обучающий. Управлением служат учебные задания, само управление производится по принципу обратной связи (см. рис.1): учебные задания вырабатываются преподавателем на основе сравнения модели требуемых знаний и модели текущих знаний студента. Последняя формируется в сознании преподавателя на основе контроля результатов выполнения студентом учебных заданий путем сравнения их с правильными результатами, которые должны быть получены преподавателем по им же выработанному заданию.

Основной задачей лабораторного практикума является обучение студента определенным понятиям, навыкам и умениям, причем это обучение в основном сводится к самостоятельной

работе студента с лабораторным практикумом. Поэтому лабораторный практикум должен выполнять функции не только некоторого учебного инструмента, но и частично функции обучающего или учителя, что привело к разработке модели обучающего. Эти функции можно поделить на два основных множества: функции обучающего – информатора и функции обучающего – оценщика (оценивающего результат выполненной работы). Кроме того, практикум должен предоставлять обучающему информацию о работе студента для возможной коррекции курса и средства для генерации новых индивидуальных заданий, обеспечивая генеративный или генеративно-диагностический подход к индивидуализации образования.

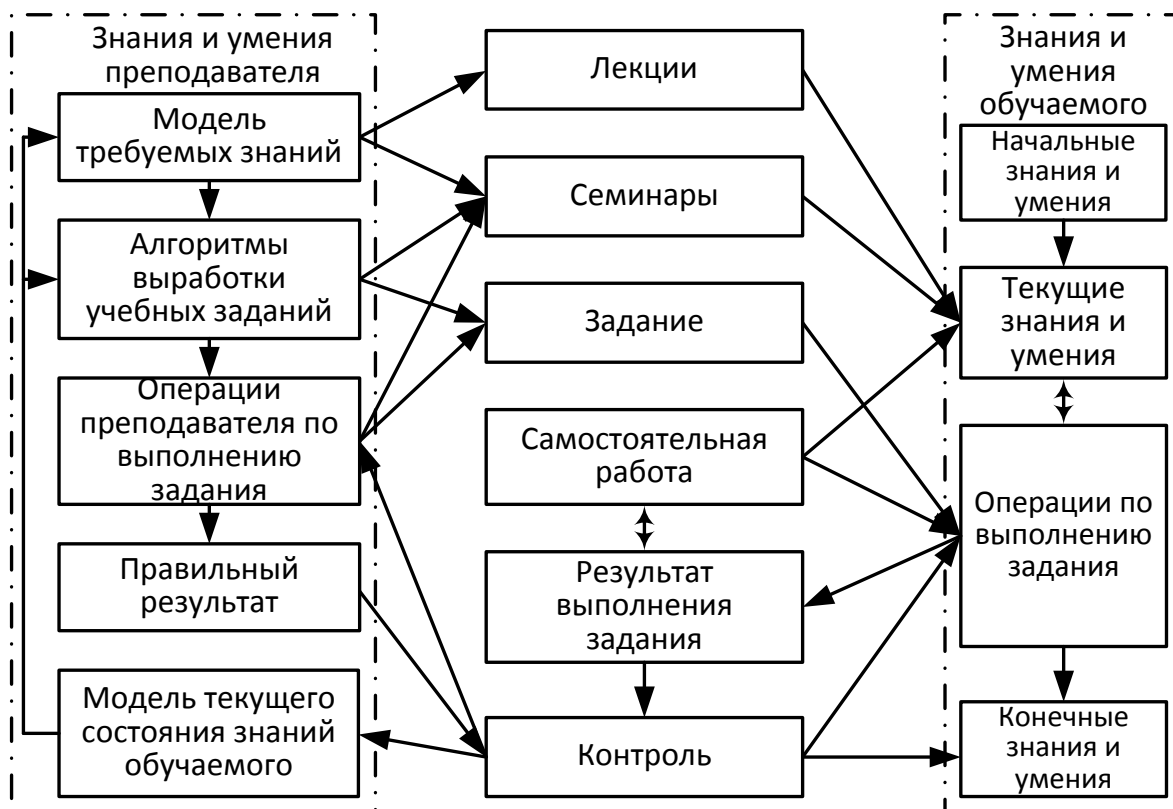


Рис.1.Схема процесса обучения

Функции обучающего – учителя заключаются в помощи выполнения работы на лабораторном практикуме. Эти функции соответственно можно поделить на помощь в работе с самим практикумом, помощь в выполнении задания и помощь в написании исходных текстов кода на встроенном языке.

Функции обучающего – оценщика заключаются в оценивании выполненной работы студента (написанного протокола заданного уровня) и выставлении некоторой оценки. Эти функции делятся на оценку качества выполненного задания и оценку работы студента.

Работа студента x над заданием k моделируются вектором характеристического профиля $E_k(x) = (p_1, \dots, p_n)$, где n – число отслеживаемых параметров, а для любого $0 < i \leq n$ p_i – значение параметра, которое собрала система о работе студента x в процессе мониторинга его работы над заданием. Для обеспечения релевантности и адекватности модели, набор параметров должен

полно описывать работу студента с комплексом. Был выделен следующий ряд параметров: время работы со справочными материалами, количество отладочных и основных запусков, время и объем редактирования исходного кода.

В процессе продвижения по курсу сдачи заданий происходит накопление данных о студенте, и вектор характеристического профиля меняется. На основе анализа временной динамики изменения вектора характеристического профиля студента делаются выводы об его усилиях. Резкое и неоправданное с позиции применяемой модели уменьшение модуля вектора говорит об использовании студентом нечестных методов решения задания.



Рис. 2. Архитектура индивидуализированных языковых инструментальных сред поддержки обучения

Для индивидуализации языковых инструментальных сред необходимо разработать семейство языков с разным синтаксисом, что дает возможность индивидуализировать среду под конкретного пользователя или предоставить каждому студенту свой вариант задания не только в части постановки задачи, но и в средствах решения. С помощью генератора-редактора правил трансляции создается индивидуальный язык описания. На основе этих правил создается индивидуальная справка по языку. Генератор заданий создает различные варианты заданий. Студент создает описание разработанной им модели на индивидуальном языке. Перед выполнением созданный студентом код скрытно от того транслируется в базовый язык, с которым среда продолжает работу. Такая трансляция позволяет не только индивидуализировать язык разработки, но и дает возможность сравнивать описания моделей, написанные на языках с разным синтаксисом, с целью оценки и выявления совпадений. В процессе работы происходит мониторинг

действий студента. Разработанная модель и результаты мониторинга передается по телекоммуникационным каналам связи для анализа и выставления учебной оценки. Общая архитектура приведена на рис. 2.

В третьей главе описывается метод создания LDC языков, созданный LDC язык упорядоченного описания взаимодействия систем, а также метод генерации языковых инструментальных сред.

Задача построения процедурного LDC языка, удовлетворяющего заданным требованиям, заключается в поиске на множестве допустимых решений такой контекстно-свободной грамматики $G = (N, T, S, P)$, на которой достигается минимум коэффициента сложности $CC = \sum_{i=1}^m \alpha_i M_i$,

$$CC = \sum_{i=1}^m \alpha_i M_i,$$

где α_i – весовой коэффициент метрики M_i сложности языка, $i = \overline{1, m}$. В решении такой задачи выделим следующие этапы:

1. Формирование целевой функции. Заключается в отборе метрик и оценке весовых коэффициентов.
2. Формирование множества допустимых грамматик. Заключается в выделении опорной составляющей, входящей во все грамматики, описывающие предметную область, и в покомпонентном ее дополнении. Обозначим через G_0 опорную составляющую, $G_0 = (N_0, T_0, S, P_0)$, где N_0 – множество нетерминальных символов, T_0 – множество терминальных символов, S – начальный символ, P_0 – продукции. Возможные описания области применения языков разобьем на группы (например, группа управления потоком команд, группа манипуляций с данными определенного типа и т.д., см. табл. 1), в каждой i -ой группе, $i = \overline{1, n}$, создадим набор из k_i компонент. Каждая j -я компонента i -ой группы состоит из множества N_{ij} нетерминальных символов, множества T_{ij} терминальных символов, множества P_{ij} продукций. Очередная допустимая грамматика G_{don} состоит из G_0 , покомпонентно дополняемой единственными представителями каждой группы, т.е.

$$G_{\text{don}} = (N_{\text{don}}, T_{\text{don}}, S, P_{\text{don}}); N_{\text{don}} = N_0 \cup \bigcup_{i=1}^n N_{ij}, T_{\text{don}} = T_0 \cup \bigcup_{i=1}^n T_{ij}, P_{\text{don}} = P_0 \cup \bigcup_{i=1}^n P_{ij}; j \in \{1, \dots, k_i\}.$$

3. Отыскание грамматики с минимальным коэффициентом сложности. Заключается в управляемом на основе метода размытых эвристик просмотре множества допустимых грамматик с подсчетом значений сформированной целевой функции.

Применим разработанный метод для синтеза языка упорядоченного описания взаимодействия систем.

Для описания статики компонентов архитектуры упорядоченного взаимодействия систем необходимо описать взаимное расположение объектов, возможные внутренние состояния объектов и возможные взаимодействия объектов и их параметры.

Взаимное расположение объектов для большей наглядности опишем в явном виде, задав разделение по уровням иерархии. Возможные взаимодействия также зададим в явном виде как список всех возможных входных событий. Внутренние состояния описываются пользователем с помощью набора переменных.

Для описания динамики компонентов архитектуры упорядоченного взаимодействия систем необходимо описать реакцию объектов на взаимодействия с другими объектами, т.е. изменение внутренних состояний и инициацию новых взаимодействий.

Каждый переход является атомарным действием, происходит либо при взаимодействии объектов с выше- и нижележащими уровнями, либо при взаимодействии внутри уровня. Описываемая архитектура основывается на слабосвязанных модулях – уровнях систем, взаимодействующих через точки доступа, с описанием примитивов взаимодействия и типов их параметров, а также событий, вызываемых по таймеру. Так как переходы по таймеру с точки зрения автомата не отличаются от переходов по входным примитивам взаимодействия, сделаем их описание единообразным. Структурная схема приведена на рис. 3.

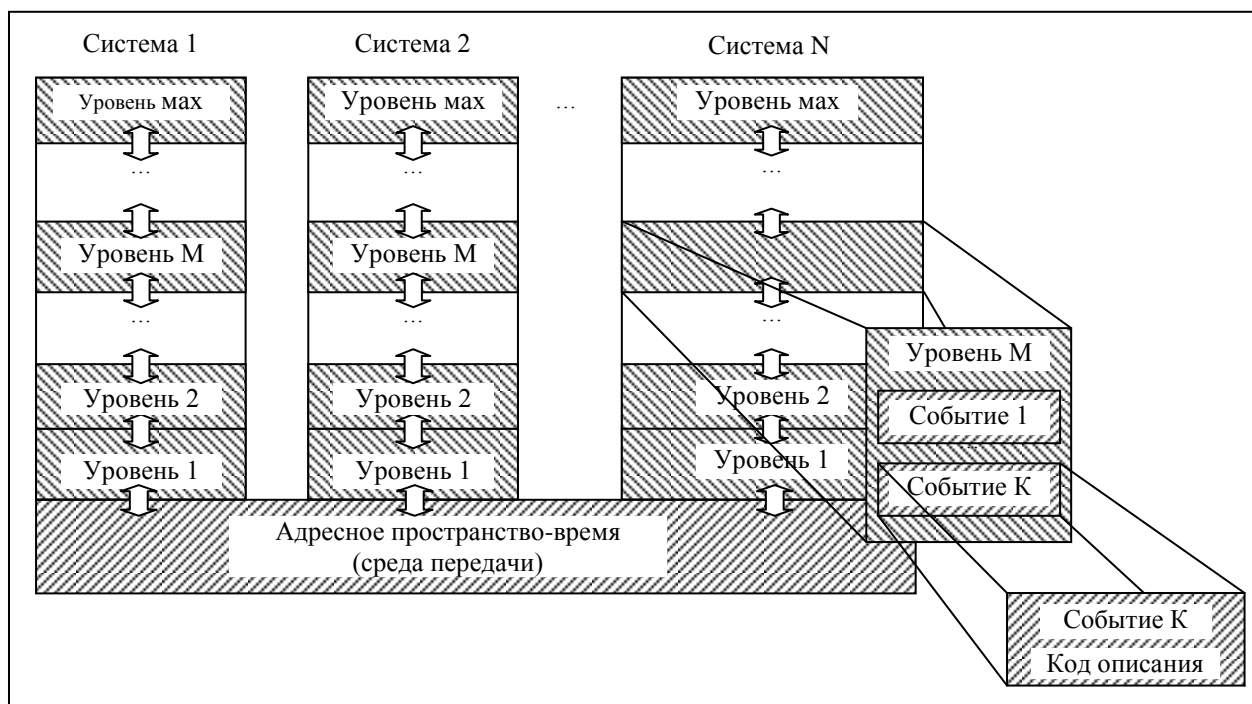


Рис. 3. Модель описания на базовом языке

Описание начинается со списка взаимодействующих систем, после которого идет список параметров среды взаимодействия. Описание системы разбивается на уровни. Внутри описания уровня находится описание возможных на нем событий – срабатывание таймера или входное

взаимодействие с соседним уровнем. Описание события содержит действия, выполняемые при возникновении события.

Составим список групп возможных описаний.

Выделим группу операторов описания взаимодействий. Введем различные операторы взаимодействия объекта с объектами выше- и нижележащих уровней. Для описания отложенных действий внутри уровня введем оператор, вызывающий заданное действие через заданный промежуток времени. Оператор для отмены неактуального отложенного действия. Оператор для определения текущего времени. Возможны следующие варианты: объединение операторов, отвечающих за взаимодействие со смежными уровнями, и операторов, отвечающих за отложенные действия, введением дополнительного параметра; замена их единственным оператором, введя несколько дополнительных параметров.

Введены три типа переменных: целочисленный – для создания всевозможных счетчиков, диапазонов таймеров и пр., а также использования в качестве логического типа; строковый – для использования в качестве конечных пользовательских данных и вывода различных диагностических сообщений; буфер (buffer) – массивы двоичных данных, в которых можно упаковывать значения нескольких переменных всех типов и которые являлись бы типом данных для передачи по сети.

Можно выделить группу формы записи операций: префиксная, постфиксная, инфиксная. Также группы системы счисления чисел и записи идентификаторов.

В группе работы с буферами возможны операторы упаковки и распаковки, шифровки/расшифровки, подсчета контрольной суммы и их комбинации.

В группе управления потоком команд выделим операторы условного и безусловного перехода, циклы с предусловием, постусловием, счетчиком и Дейкстры.

В группе работы с очередями используются оператор добавления элемента в очередь и функции извлечения элемента из очереди и получения количества элементов в очереди. Дополнительно различные формы организации.

В группе манипуляций со строками операторы извлечения подстроки и удаления подстроки, функция нахождения подстроки и конкатенация.

Для доказательства полноты языка на нем была реализована машина Тьюринга.

Для оценки сложности языка воспользуемся следующими метриками: число нетерминалов, сложность Маккейба, средний размер продукции. Результаты оценки и сравнение с рядом популярных языков приведены в табл. 1.

Для индивидуализации инструментальных средств, позволяющих подстроить среду под конкретного пользователя необходимо создать инструмент, позволяющий модифицировать язык разработки (вводить пользовательский язык). Перед выполнением созданный пользовате-

лем код скрытно от того транслируется в базовый язык, с которым среда продолжает работу. Такая трансляция позволяет не только индивидуализировать язык разработки, но и упростить среду – вся работа с кодом выполняется в синтаксисе базового языка.

Таблица 1. Оценка сложности различных языков

Метрика размера/сложности	Разработанный язык	E-LOTOS	SDL-92	ISO C	ISO C++	Java
Число нетерминалов	36	34	119	64	141	149
Сложность Маккейба	64	277	378	149	368	213
Средний размер	5,6	22,1	4,6	7,6	6,1	4,1

Так как естественный образ записи выражений в абсолютном большинстве случаев изменять было признано нецелесообразным, то модификация синтаксисов коснулась только операторов языка. В связи со строчной структурой базового языка было принято решение проводить построчную трансляцию, при этом нахождение операторов базового языка в начале строки также облегчало задачу. Работа транслятора распадается на две фазы:

1. Деление исходной строки на токены, каждый из которого представляет идентификатор, например, имя оператора или переменной, либо выражение.

2. Поиск среди токенов имени оператора, и добавление в исходящую строку имени, которое соответствует имени оператора в базовом синтаксисе. Помимо этого, выполняется перестановка параметров оператора по правилам, которые определяются из внешнего источника. Трактовкой правил внешнего источника занимается простой интерпретатор. Эти правила устанавливают семантическое соответствие между конструкциями базового и пользовательского языков и описываются языком на основе грамматики G_{mp} транслятора.

В итоге для порождения пользовательского языка требуется решить следующую задачу: построить генератор Γ такой, что $\Gamma^{-1}: P_{tr}(G_i) = G_b$, где G_b – грамматика базового языка, G_i – грамматика индивидуального языка, P_{tr} – правила работы транслятора.

Общая схема работы генератора представлена на рис. 4. На основе алфавита терминальных символов базового языка T_b создается новый алфавит терминальных символов индивидуального языка T_i , основанный на наглядности сочетаний алфавита, ключевых слов и спецсимволов. Далее происходит подстановка нового алфавита в грамматику базового языка G_b . В заключительной фазе путем рекурсивного (в связи с обратным характером задачи) случайного или заданного применения обратных правил работы транслятора формируется грамматика индивидуального языка G_i и набор правил P_{tr} для транслятора из G_i в G_b .

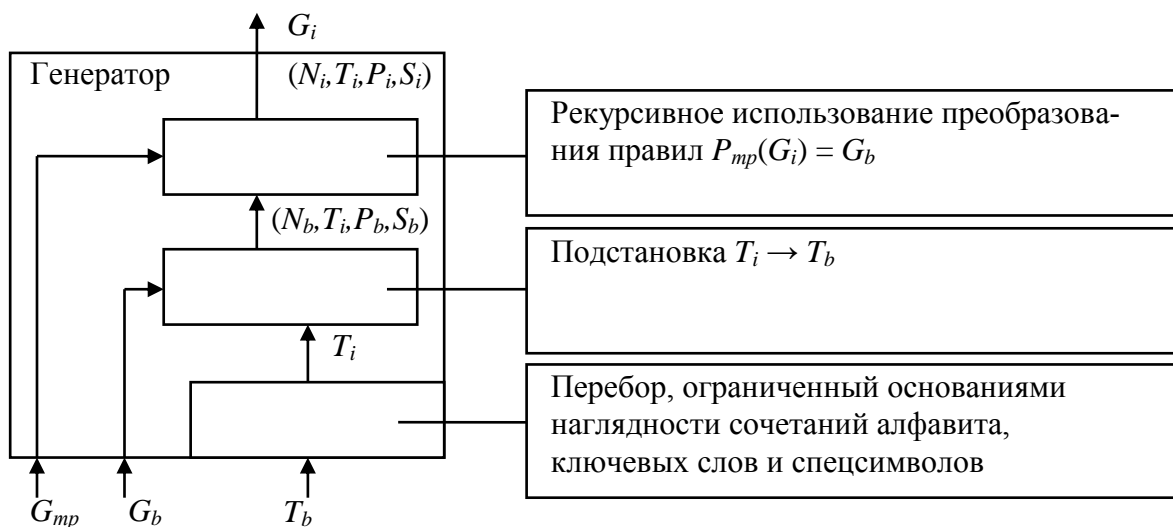


Рис. 4. Схема работы генератора синтаксисов индивидуальных языков

В четвертой главе описывается разработанный программный комплекс поддержки лабораторных занятий. Структура комплекса представлена на рис. 5.



Рис.5. Структура лабораторного комплекса

Основные функции комплекса:

- Предоставление студенту справочной системы, включающей в себя справочную информацию об использовании комплекса и методические указания по выполнению задания.

- Предоставление студенту индивидуализированной среды поддержки разработки протоколов, включающей в себя язык их описания (с интерпретатором), редактор и тестирующие средства.
- Предоставление обучающему средств оценки выполненного задания.
- Предоставление обучающему средств определения нечестного выполнения задания.
- Обеспечение передачи выполненного задания по телекоммуникационным каналам.
- Предоставление средств разработки новых заданий.

Справочная система предоставляет студенту справку по пользовательскому интерфейсу комплекса, индивидуальную справку по синтаксису языка его задания и методические указания по выполнению задания. Кроме того, справочная система сохраняет данные о частоте и продолжительности ее использования. Для упрощения создания справки по пользовательскому варианту языка создан *инструмент генерации справки*. На основе шаблонов создается комплект файлов, готовых к компиляции в справочную систему. Строки названия, синтаксиса и примера генерируются на основе описания базового языка и правил, которыми руководствуется транслятор из индивидуального языка в базовый. При изменении синтаксиса индивидуального языка новая справка генерируется в автоматическом режиме.

Среда поддержки разработки протоколов состоит из: системного исполнителя, эмулятора сетевого уровня, интерпретатора встроенного языка, транслятора из индивидуального языка в базовый, встроенного отладчика, анализаторов статистики, набора редакторов. Системный исполнитель хранит данные и связывает основные компоненты практикума – эмулятор сетевого уровня и интерпретатор встроенного языка. Эмулятор сетевого уровня обеспечивает сервис сетевого уровня семиуровневой модели ВОС. Он способен менять свое поведение в зависимости от задаваемых показателей качества, отражающих показатели качества реальных сетевых сред, и предоставляет для использования студентом сетевой сервис, содержащий набор услуг, достаточный для целей обучения в рамках домашнего задания. Интерпретатор производит обработка кода обработчиков событий, написанных на базовом языке. Разрабатываемый студентом протокол на языке с индивидуальным синтаксисом транслируется в язык базового синтаксиса и выполняется интерпретатором, системным исполнителем и эмулятором сетевого уровня. Правила работы транслятора хранятся во внешнем файле, что позволяет легко изменять или добавлять синтаксисы пользовательских языков без изменения исполняемого файла. Встроенный отладчик помогает отлаживать протоколы, а анализаторы статистики проводят оценку протокола.

Оценка выполненного задания производится как на основе оценки качества самого протокола, так и характеристик работы студента над ним. Первым этапом оценки выполненной работы является проверка качества полученного сервиса на соответствие требованиям задания. Проверка выполняется с помощью специальных тестеров, написанных на встроенном языке и соби-

рающих необходимую статистику. Если результаты не укладываются в установленные требованиями пределы, то работа отправляется на доработку. Если же полученный сервис соответствует требованиям задания, то производится оценка как качества самого протокола, так и характеристик работы студента над ним.

Качество протокола оценивается сравнением характеристик протокола («взвешенный» размер протокола и «взвешенные» потребляемые ресурсы при работе протокола) с характеристиками уже сданных заданий того же варианта. В случае успешной сдачи задания эти характеристики заносятся в базу данных, сдвигая границы оценки и приближая их к характеристикам протокола «среднего студента».

Работа студента над заданием определяется на основе анализа ряда параметров: времени работы со справочными материалами, количества отладочных и основных запусков, времени и объемом редактирования элементов протокола. Для повышения надежности распознавания нечестных методов применяются дополнительные механизмы: отслеживание исходного варианта и авторства выполняемого задания и сравнение сдаваемого задания с уже сданными и занесенными в базу данных. Эти средства определения нечестного выполнения задания дают возможность обучающему оценить степень самостоятельности работы студента.

Телекоммуникационный модуль обеспечивает передачу выполненного задания от студента к обучающему по телекоммуникационным каналам связи. Была выбрана передача с помощью средств электронной почты, как одного из наиболее доступного и дешевого средства связи.

Для автоматизации подготовки вариантов заданий был разработан *инструмент генерации текстов задания*. Для каждого задания создается шаблон или несколько шаблонов, если варианты значительно различаются. Создается таблица параметров заданий, в которой хранится соответствие между номером варианта и параметрами задания (включая номер шаблона). При необходимости генератор создает заданный вариант путем подстановки параметров в шаблон. Этот способ удобен тем, что все различия вариантов можно представить в удобном табличном виде, и они легко редактируются. Путем редактирования шаблона можно легко изменять задания. Для облегчения разработки индивидуальных синтаксисов был создан редактор, позволяющий в наглядной форме редактировать правила трансляции персональных синтаксисов в базовый. Кроме того, этот редактор имеет в своем составе генератор подобных правил.

Кроме того, для поддержки комплекса были разработаны сайт, на котором студенты могли получить новости и обновления программных компонент комплекса, и конференция, на которой студенты могли общаться и получать ответы на вопросы по выполнению заданий и техническую поддержку.

В пятой главе приводятся данные экспериментов по использованию разработанных инструментов и методов генерации индивидуальных сред и оценки выполненных заданий, критерии разработки заданий.

Для экспериментальной проверки система была внедрена в учебный процесс кафедры «Кибернетика» МИФИ для поддержки лабораторных работ по курсу «Взаимосвязь открытых систем». Внедрение позволило перевести лабораторные занятия в форму домашнего задания с возможностью дистанционной сдачи. Первоначальный вариант комплекса не включал в себя средств индивидуализации и построения профиля студента. В результате апробации было решено расширить его системой построения характеристического профиля студента. Это дало значительное увеличение степени достоверности данных об индивидуальной работе каждого студента над выполнением заданий, что позволило более справедливо оценивать их работы. Следующим шагом было принятие решения о введении системы, стимулирующей самостоятельность выполнения задания, что привело к появлению системы индивидуализации средств разработки. Дальнейшее использование комплекса подтвердило эффективность разработанных средств. Количество обнаруженных совпадений в сдаваемых заданиях представлено на рис. 5. На этом рисунке видно значительное снижение количества списанных работ после введения системы индивидуализации в 2006 году.

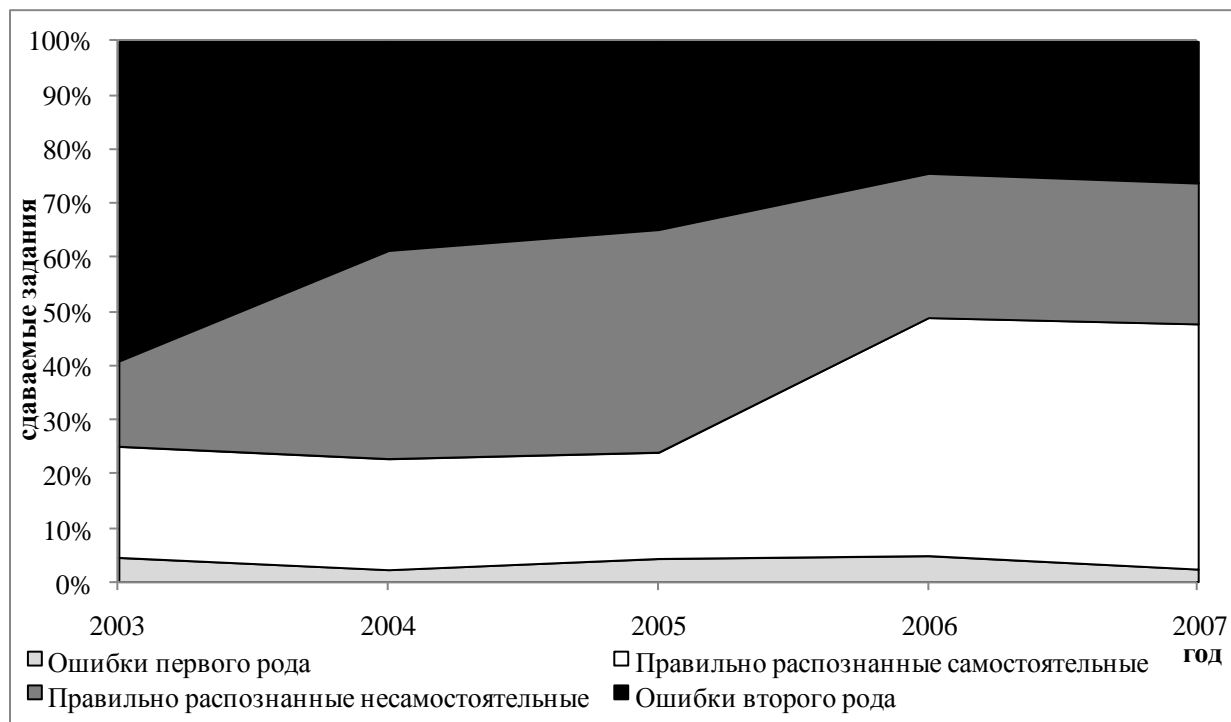


Рис. 5. Количество обнаруженных совпадений в сдаваемых заданиях по годам

Созданный язык и инструментальная среда поддержки разработки были использованы в рамках проекта "ИнфраМенеджер" ЗАО «Софтинтегро». Их использование позволило сокра-

тить время цикла разработки элемента пользователя, обеспечивающего сопряжения моделей приложений и стека протоколов.

В заключении отражены основные результаты, полученные в данной работе.

В приложении содержатся акты о внедрении и использовании результатов диссертационного исследования.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Основные результаты, полученные автором диссертационного исследования, состоят в следующем:

1. Проанализированы существующие подходы к оценке сложности языков. В результате анализа предложен класс LDC языков.
2. Разработана новая архитектура индивидуализированных языковых инструментальных сред поддержки обучения, основанная на индивидуализации среды под пользователя.
3. Разработан новый метод создания LDC языков, позволяющий создавать языки низкой описательной сложности, обеспечивающей простоту их освоения и реализации и поддержки основанного на них программного обеспечения.
4. Создан LDC язык упорядоченного описания взаимодействия систем, позволяющий сократить время на освоение использующих его средств.
5. Создан новый метод генерации языковых инструментальных сред, позволяющий индивидуализировать среду под пользователя.
6. Разработана среда моделирования упорядоченного взаимодействия распределенных систем.
7. На основе предложенных в диссертации моделей и методов разработаны инструментальные программные средства генерации языковых инструментальных сред.
8. Проведена апробация предложенных в диссертации моделей и методов на базе реализованного программного комплекса поддержки лабораторных занятий по курсу «Взаимосвязь открытых систем», которая подтвердила работоспособность разработанного метода генерации языковых инструментальных сред.
9. Проведено полномасштабное внедрение разработанного программного комплекса поддержки лабораторных занятий по курсу «Взаимосвязь открытых систем» в учебный процесс МИФИ.
10. Разработанный LDC язык упорядоченного описания взаимодействия систем использован в проекте "ИнфраМенеджер" ЗАО «Софтинтегро».

Результаты работы показывают, что поставленная цель создания инструментальных средств синтеза языков и протоколов взаимодействия распределенных систем можно считать достигнутой.

Основные положения диссертационной работы опубликованы в печатных работах [1-10]. Положения диссертации отражены также в учебном пособии [11].

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Никифоров А.Ю. Язык описания взаимодействия иерархических систем и его персонализация. // Программные продукты и системы. – 2009. – №1 – С. 36-37.
2. Никифоров А.Ю., Русаков В.А. Комплекс поддержки разработки учебных вариантов протоколов // Научная сессия МИФИ-2001: Сборник научных трудов. – Т.3. – М.: МИФИ, 2001. – С. 226-227.
3. Никифоров А.Ю. Комплекс поддержки разработки учебных вариантов протоколов // Научная сессия МИФИ-2002: Сборник научных трудов. – Т.2. – М.: МИФИ, 2002. – С. 61-62.
4. Никифоров А.Ю., Русаков В.А. Проблемы аутентичности и формирования профиля обучаемого в программном комплексе поддержки лабораторных занятий // Научная сессия МИФИ-2003: Сборник научных трудов. – Т.2. – М.: МИФИ, 2003. – С. 66-67.
5. Никифоров А.Ю., Русаков В.А. Формирование профиля обучаемого в программном комплексе поддержки лабораторных занятий // Научная сессия МИФИ-2004: Сборник научных трудов. – Т.2. – М.: МИФИ, 2004. – С. 126-127.
6. Никифоров А.Ю., Русаков В.А. Модель студента в программном комплексе поддержки лабораторных занятий // Научная сессия МИФИ-2005: Сборник научных трудов. – Т.2. – М.: МИФИ, 2005. – С. 79-80.
7. Никифоров А.Ю., Русаков В.А. Разработка модели обучаемого в программном комплексе поддержки лабораторных занятий // Научная сессия МИФИ-2007: Сборник научных трудов. – Т.2. – М.: МИФИ, 2007. – С. 46-47.
8. Никифоров А.Ю. Виртуальная индивидуализация среды поддержки дистанционного обучения. // Современные технологии в задачах управления, автоматизации и обработки информации: труды XVI международного научно-технического семинара. Сентябрь 2007г., Алушта. – Тула: Изд-во ТулГУ, 2007. – С. 163-164.
9. Никифоров А.Ю. Индивидуализация среды поддержки обучения. // XII московская международная телекоммуникационная конференция студентов и молодых ученых "Молодежь и наука". – М.: МИФИ, 2009. – С. 117-118.
10. Никифоров А.Ю. Индивидуализация языков описания. // XV международная научно-техническая конференция студентов и аспирантов "РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА". – Т. 1. – М: Издательский дом МЭИ, 2009 – С. 312-313.
11. Никифоров А.Ю., Русаков В.А. Взаимосвязь открытых систем (основы теории и практики). Учебное пособие. – М.: МИФИ, 2005. – 92 с.