

РОССИЙСКАЯ АКАДЕМИЯ НАУК

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

им. А.А. Дородницына

Н.В. Белотелов, Ю.И. Бродский, Ю.Н. Павловский

**Компьютерное моделирование
демографических, миграционных,
эколого-экономических процессов
средствами распределенных
вычислений**

**ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР ИМ. А.А. ДОРОДНИЦЫНА РАН
МОСКВА 2008**

УДК 519.87

Ответственный редактор
чл. – корр. РАН Ю.А. Флеров

Монография посвящена проектированию инструментальной системы распределенного имитационного моделирования. К обсуждаемым в ней вопросам относятся выбор класса моделей и их компонент, на которые будет ориентирована проектируемая инструментальная система моделирования; выбор архитектуры инструментальной системы; способы взаимодействия распределенных частей модели; концепция моделирования, предлагаемая исследователю инструментальной системой моделирования и, наконец, набор средств, предоставляемых проектируемой системой.

В работе рассматриваются вопросы, связанные с использованием инструментальной системы распределенных вычислений при моделировании демографических и миграционных процессов.

Ключевые слова:

распределенное имитационное моделирование, демографические и миграционные процессы, распределенные вычисления.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант 07-07-00071-а.

Рецензенты: А.А.Петров,
А.А. Васин

Научное издание

© Вычислительный центр им. А.А. Дородницына
Российской академии наук, 2008

Введение

Основная цель настоящей публикации состоит в том, чтобы с содержательной точки зрения обосновать перспективность использования технологии распределенных вычислений для изучения сложных процессов с помощью компьютерного моделирования. Для иллюстрации такой перспективности используется эколого-демографо-экономическая модель (ЭДЭМ). Эта модель была разработана в течение 2001-2003 гг. в Вычислительном центре РАН им. А.А. Дородницына по заказу Института химии и проблем устойчивого развития Российского химико-технологического университета им. Д.И. Менделеева [1-3]. ЭДЭМ описывает демографические, экономические, экологические процессы в системе регионов. С 2004 г. компьютерная реализация варианта этой модели эксплуатируется в учебном процессе в РХТУ, МФТИ, ВМК МГУ для иллюстрации студентам некоторых аспектов проблем устойчивого развития.

ЭДЭМ можно считать одним из развитий модели «Мировая динамика» (МД) Дж. Форрестера [4]. Широко известные результаты этой модели таковы. Если количество невозобновляемых природных ресурсов на единицу производимого продукта в течение ближайших нескольких десятков лет не уменьшится существенно, то рост численности человечества остановится по причине увеличения расходов на извлечение природных ресурсов, уменьшения эффективности производственных фондов, сокращения производства продуктов питания, увеличения из-за этого коэффициента смертности.

Если же развитие технологий справится с проблемой исчерпания природных ресурсов, и описанный механизм ограничения численности людей не будет действовать, то несколь-

ко позднее включится другой механизм ограничения численности людей, связанный с увеличением загрязнения, сокращения вследствие этого рождаемости и увеличения смертности.

Если же «исключить» и этот механизм, то «следующим» механизмом ограничения численности людей на Земле будет ее «перенаселенность», влекущая сокращение площади сельскохозяйугодий, производства продуктов питания, рождаемости, и увеличение смертности.

Конечно, ничего такого, что предсказывает модель Дж. Форрестера, в реальности не произойдет. В частности, по некоторым данным скорость прироста численности людей на Земле в настоящее время замедляется. В [5-6] это обстоятельство названо «демографическим переходом». Высказываются предположения о механизмах этого явления. Однако ничего общего с механизмами уменьшения численности людей на Земле, имеющими место в МД, они не имеют. Скорее всего, это замедление определяется возрастающим уровнем образованности женщин, вовлечением их в производственные процессы и увеличением их социальных притязаний. МД была усовершенствована в работах [7-8]. Характер этих моделей по отношению к проблеме их идентификации и адекватности таков же, как и характер модели МД.

Вернемся к модели ЭДЭМ. Если ЭДЭМ используется для воспроизведения процессов в масштабе всей Земли, то по отношению к проблеме ее идентификации и адекватности результатов характер ЭДЭМ такой же, каков характер модели МД. Именно, так же, как МД, ЭДЭМ, при использовании для всей Земли, в настоящее время невозможно идентифицировать (за исключением модели демографического процесса), поскольку необходимые для этого характеристики не измеряются в эконометрике. Другими словами, ЭДЭМ в таком мас-

штабе «не наблюдаема» относительно имеющейся системы измерений. Поэтому внешние характеристики в ЭДЭМ, так же как и в МД подбираются так, чтобы получающиеся результаты допускали осмысленную интерпретацию.

Однако по мере уменьшения масштабов моделирования ситуация с идентификацией ЭДЭМ улучшается. В самом деле, что такое уровень загрязнения всей Земли в целом и как его измерить, не очень ясно. Конечно, можно оперировать такими глобальными характеристиками, как концентрация CO_2 или процент ненарушенных экосистем [9]. Но описать влияние этих характеристик на рождаемость, смертность, экономические характеристики не представляется возможным. В то же время уровень загрязнения региона, являющегося, например, субъектом Российской Федерации в ряде случаев вполне может быть измерен. Вообще чем меньше регион, тем проще задача описания уровня его загрязнения. То же касается производственных фондов и природных ресурсов. При этом, однако, в разных регионах загрязнение будет разным по своему характеру, что может создать определенные трудности при расчетах усредненных по регионам характеристик.

В настоящее время во многих регионах России ведутся работы по компьютерному моделированию демографических, экономических, экологических процессов [10-12]. В каждом из этих исследований выделяются те аспекты указанных процессов, которые представляются разработчикам главными для данного региона. В большинстве случаев эти работы практически не связаны друг с другом. При этом используются разные концепции моделирования, разные инструментальные средства. Между тем регионы России не являются замкнутыми и взаимосвязи между ними весьма существенны. Характеристики взаимосвязей между регионами в региональных моделях

являются внешними для каждой такой модели и должны конкретизироваться при выполнении расчетов.

Представляется естественным разработать «типовую» модель региона, как это имеет место в США [13]. К компьютерной реализации такой модели в каждом конкретном регионе следует предъявить требование способности функционировать в режиме распределенных вычислений. Это означает возможность запускать модель с отдаленных компьютеров, получать на эти компьютеры результаты моделирования и обрабатывать эти результаты в соответствии с потребностями того, кто запускает с отдаленного компьютера региональную модель. Это даст возможность, например, при анализе и прогнозе социально-экономических характеристик в некотором регионе вместо конкретизации внешних величин, зависящих от соседних (других) регионов, обращаться с помощью режима распределенных вычислений к моделям этих регионов. Не исключена ситуация столь тесной связи между регионами, что вычисление их характеристик по их замкнутым моделям не будет давать адекватных результатов. Наконец, возможность выполнять расчеты по моделям регионов, предоставляемая режимом распределенных вычислений, дает федеральным структурам дополнительные механизмы анализа и прогноза последствий принимаемых на федеральном уровне решений, касающихся регионов, а также дополнительные возможности контроля за их деятельностью.

Естественно начать такую работу с моделей демографических процессов. Во-первых, с математической точки зрения модели демографических процессов наиболее просты. Во-вторых, они легче других поддаются идентификации. В третьих, демографические процессы являются предметом постоянного интереса, как администрации регионов, так и федеральных властей. В четвертых, предстоящая перепись населения

России в 2010 г. делает задачу прогнозирования демографических процессов в регионах важной, актуальной и интересной.

Настоящая публикация состоит из четырех разделов. В первом разделе излагается так называемый «базовый» вариант ЭДЭМ. (слово «базовый» поясняется далее) и дается системный анализ проблемы устойчивого развития. Словосочетание «системный анализ» означает здесь гуманитарный анализ проблемы устойчивого развития, использующий, однако, понятия и положения, возникшие при построении модели и выполнении расчетов по модели.

Во втором разделе излагается проект моделей демографических процессов в системе регионов. При использовании этих моделей становится возможным воспроизводить в режиме распределенных вычислений демографический процесс в системе регионов, в которой эти модели реализованы с учетом миграционных процессов между регионами. Это означает, что некоторый пользователь со своего компьютера может воспроизвести демографический процесс в системе тех регионов, где на региональных компьютерах имеются реализации модели демографического процесса. При этом в каждом регионе будет запущена на региональном компьютере модель демографического процесса в этом регионе. Результаты поступят на компьютер пользователя, который может обрабатывать их как ему необходимо.

В третьем разделе приведена концепция построения инструментальной системы распределенного моделирования.

В четвертом разделе кратко описывается имеющаяся распределенная реализация ЭДЭМ.

В заключении проводится обсуждение некоторых результатов моделирования и рассматриваются различные аспекты проблем устойчивого развития.

1. Эколого – демографо - экономическая модель

Описание базовой модели

В настоящем разделе дается словесное (без формул) описание базового варианта эколого – демографо – экономической имитационной модели (ЭДЭМ) [1-3]. Термин «базовая модель» означает следующее. При разработке имитационной модели сложного процесса рационально на первом этапе разработки не учитывать ограничения на размерность модели и ее сложность, связанные с необходимостью ее идентификации, и компьютерной реализации. Такая модель и называется «базовой». Она является корнем дерева моделей, получаемых далее из нее упрощениями различного характера, которые зависят от целей исследования, от имеющейся техники, трудовых ресурсов, возможностей идентификации.

Базовая модель воспроизводит демографические, экономические и экологические процессы на характерных временах, соизмеримых со временем жизни поколения.

Модель демографического процесса

Модель воспроизводит эволюцию половой и возрастной структуры населения в странах (регионах). Внутренними характеристиками являются количества в данном году женщин и мужчин, имеющих данный возраст (от 0 до 100).

Внешними характеристиками являются распределение населения в стране по полу и возрасту в начальном году, т. е. в году, с которого начинается модельное воспроизведение демографического процесса, а также зависимости коэффициентов рождаемости и смертности от возраста, образованности,

экологических факторов, от состояния системы здравоохранения. Для каждой страны по распределению населения по полу и возрасту в данном году рассчитывается это распределение в следующем году.

Прогностическими характеристиками, кроме упомянутых выше, также являются:

- общая численность населения в стране,
- общая численность мужчин,
- общая численность женщин,
- количества родившихся и умерших в данном году,
- численность детей (возраст от 0-го года до 6),
- численность школьников (возраст от 7 лет до 17),
- численность студентов (возраст от 18 лет года до 22; эта величина определяется по расходам государства и домашних хозяйств на высшее образование — см. разд. 4.3),
- численность аспирантов (возраст от 22 лет года до 24; эта величина определяется по расходам государства и домашних хозяйств на послевузовское образование --- см. разд. 4.3),
- численность рабочих (возраст от 18 лет до 59 — рабочими считаются мужчины и женщины, окончившие школу, но не получившие высшего образования),
- численность служащих (возраст от 23 лет до 59 — служащими считаются мужчины и женщины, получившие высшее образование, но не окончившие аспирантуру),
- численность ученых (возраст от 25 лет до 59 — учеными считаются мужчины и женщины, окончившие аспирантуру),
- численность активного населения, т. е. суммарная численность рабочих, служащих, ученых,

- численность пенсионеров,
- отношение численности пенсионеров к численности активного населения и т.д.

Виды экономической деятельности, учитываемые в эколого-социально-экономической имитационной модели

В основе экономической части ЭДЭМ лежит работа [2]. Учитываются следующие виды экономической деятельности:

- добыча природных ресурсов,
- производство предметов потребления,
- образование,
- здравоохранение,
- экологическая деятельность,
- информатика (сбор, хранение, обработка, передача информации и производство инструментов, обеспечивающих эти виды деятельности),
- производство новых знаний (фундаментальная наука),
- производство новых технологий (прикладная наука),
- производство вооружений,
- производство того, что принято называть «духовными ценностями» (религия, культура, искусство).

В целях краткости изложения и допуская вольность речи, все отмеченные выше виды деятельности будут называться здесь «отраслями производства», а производство духовных ценностей будет для краткости называться словом «культура». В ЭДЭМ возможно также появление в процессе функционирования производственной структуры новых отраслей производства (видов деятельности), производящих новые пред-

меты потребления, которых не существовало в момент времени, соответствующий началу процесса имитации.

Добывающие отрасли извлекают природные ресурсы и подготавливают их для использования в других отраслях. Обрабатывающие отрасли производят «традиционные» предметы потребления. Образовательная отрасль производит «образованность» людей, необходимую при осуществлении производственных процессов во всех других отраслях в соответствии с действующими в них технологиями (см. далее описание технологий).

Здравоохранение «производит» коэффициенты человеческой рождаемости и смертности (более точно — зависимости этих коэффициентов от возраста, образованности, качества жизни и параметров, характеризующих степень загрязнения среды обитания).

Экологическая отрасль собирает информацию о загрязнителях, выбрасываемых в окружающую среду в процессе производства, изучает их влияние на биосферу и здоровье людей. Эти знания, наряду со знаниями, которые добываются фундаментальной наукой (см. ниже), являются «сырьем» для разработки новых технологий, уменьшающих выбросы загрязнений в окружающую среду (см. ниже описание технологий). Таким образом, можно считать, что экологическая отрасль производит параметры окружающей среды, обеспечивающие комфортное существование людей.

Информационная отрасль производит средства сбора, обработки, хранения, передачи информации и саму информацию, являющуюся в ЭДЭМ как предметом конечного потребления людей, так и продуктом, используемым в процессе производства во всех других отраслях [14,15].

Фундаментальная наука производит новые знания о природе. Новые знания используются прикладной наукой для производства новых технологий.

В рамках ЭДЭМ никакие вооруженные конфликты между странами имитировать не предполагается, т. е. заранее считается, что виртуальный мир, воспроизводимый ЭДЭМ, развивается без вооруженных конфликтов. Поэтому можно считать, что оборонные отрасли стран производят их «безопасность». Другими словами, оборонные отрасли производят «отсутствие в моделируемом мире вооруженных конфликтов».

Технологии в ЭДЭМ

В одной и той же отрасли могут использоваться несколько технологий. Технологии в ЭДЭМ характеризуются, во-первых, количествами продуктов, вообще говоря, всех других отраслей производства, необходимых для обеспечения выпуска единицы продукта по данной технологии в данной отрасли (леонтьевские коэффициенты). Поскольку образование (обучение людей) в ЭДЭМ считается отраслью производства, то в эту систему величин входят трудозатраты людей различного уровня образованности, необходимые для производства единицы продукции данной отрасли. Таким образом, указанные трудозатраты являются в ЭДЭМ леонтьевскими коэффициентами.

Во-вторых, технологии в ЭДЭМ характеризуются количеством природных ресурсов, необходимых для производства единицы продукции.

В-третьих, технологии характеризуются количеством загрязнений, выбрасываемых в окружающую среду на единицу производимого продукта.

В-четвертых, технологии характеризуются количеством продуктов всех других отраслей (в том числе трудозатрат), необходимых для того, чтобы построить единицу производственной мощности по данной технологии.

В-пятых технологии характеризуются трудозатратами, необходимыми для построения единицы мощности.

Каждая технология в каждой стране характеризуется распределением мощности (максимального выпуска продукции) по возрасту (или, что то же самое, по времени ввода в строй этой мощности).

Для того чтобы появилась мощность, соответствующая данной технологии, необходимо, во-первых, чтобы сама эта технология была произведена прикладной наукой, во-вторых, чтобы в некоторый момент времени в эту технологию частными фирмами (см. ниже) были произведены инвестиции. Момент появления мощности отстоит от момента инвестиций на «время строительства», которое является внешней характеристикой модели.

Экономические агенты в ЭДЭМ

Экономическими агентами называются субъекты, участвующие в процессах производства, распределения, обмена, потребления материальных благ. В ЭДЭМ учитываются три типа экономических агентов: правительства стран, частные фирмы и домашние хозяйства.

Правительства имеют средства в результате сбора налогов с прибылей фирм (налог на прибыль), сбора налогов на выбросы загрязнителей (экологический налог) и с получаемой в домашних хозяйствах зарплаты (подходный налог). Правительство может направлять эти средства на развитие образо-

вания, науки, на здравоохранение, культуру, на пенсионное обеспечение, выплаты безработным, на оборону.

Правительство, в принципе, может производить прямые инвестиции в различные отрасли экономики. Однако государственная собственность в ЭДЭМ отсутствует и соответствующие мощности после ввода их в строй будут считаться принадлежащими частным фирмам. Ставки налогообложения и распределение расходов по указанным статьям являются государственным управлением (т. е. внешними характеристиками, задаваемыми в рамках сценариев имитационных экспериментов или имитационных игр).

Частные фирмы инвестируют в отрасли, а в пределах отрасли — в технологии, приносящие прибыли. Механизм, в силу которого инвестиции распределяются по имеющимся и новым технологиям, конкретизируется в рамках сценария имитационного эксперимента. Новые технологии производит прикладная наука, используя добытые наукой новые знания.

Домашние хозяйства тратят получаемую заработную плату в соответствии со *структурой потребления*. Структура потребления в ЭДЭМ — это доли истраченных домашними хозяйствами заработанных продаж своего труда на рынке труда средств (т. е. зарплаты) на приобретение продукции отраслей производства (в том числе средства, истраченные на воспитание детей, образование, здравоохранение, продукцию информационной отрасли, культуру). Указанные доли зависят от многих факторов.

Модель производственных процессов

Основной характеристикой производственных процессов в странах является распределение производственных мощно-

стей по времени, эксплуатационному возрасту, отраслям экономики и технологиям. Значение этой характеристики есть максимально возможный выпуск продукции в единицу времени (т. е. в год).

Для каждой страны в каждом году для каждого эксплуатационного возраста каждой технологии в каждой отрасли вычисляется рентабельность этой мощности. Для этого от рыночной стоимости единицы продукции данной отрасли (см. ниже описание модели рынка) вычитаются затраты, которые необходимо осуществить для того, чтобы эту единицу продукции произвести.

Затраты состоят из стоимости продукции всех других, вообще говоря, отраслей, необходимых для производства единицы продукции по данной технологии (см. выше описание технологий), из рыночной стоимости труда различного уровня образованности, необходимого для производства единицы продукции по данной технологии, из стоимости эксплуатации данной мощности, зависящей от ее эксплуатационного возраста, а также из выплат за выброс загрязнителей в расчете на единицу выпуска (см. выше). Если рыночная стоимость единицы продукции, произведенной данной мощностью, превышает затраты, то данная мощность считается рентабельной.

Рентабельные мощности предъявляют спрос на труд различного уровня образованности на рынок труда данной страны (см. ниже описание модели рынка труда). Предложение труда на рынок труда в данной стране вычисляется в демографическом блоке (см. выше).

Таким образом, для каждой страны и каждой отрасли вычисляются спрос и предложение труда данного уровня образованности. Это дает возможность вычислить цену труда в следующем году в соответствии с моделью рынка труда, а

также определить те рентабельные мощности, которые обеспечены рабочей силой.

Если рабочей силой (в данной отрасли) обеспечены все рентабельные мощности, то это означает, что в данной отрасли имеет место безработица, уровень которой в этом случае определяется. В этом случае рабочая сила ранжируется по степени квалификации и рабочие места получают начиная с самых квалифицированных и далее до исчерпания рентабельных мощностей. Принцип ранжирования по степени квалификации в базовой модели не конкретизируется. Если рабочей силы для обеспечения всех рентабельных мощностей недостаточно, то мощности ранжируются по уровню рентабельности и рабочей силой обеспечиваются начиная с наиболее рентабельных и далее насколько хватит рабочей силы. Таким образом определяется выпуск продукции в каждой отрасли в каждой стране, а также производственные затраты продукции всех отраслей, обеспечивающие этот выпуск.

Далее вычисляется потребление продукции каждой отрасли в каждой стране. Продукцию отраслей производства потребляют домашние хозяйства и частные фирмы. Потребление продукции домашними хозяйствами характеризуется *структурой потребления*. Структура потребления — это доля заработной платы, которая тратится домашними хозяйствами на приобретение продукции различных отраслей производства. В ЭДЭМ структура потребления является внешней характеристикой, имеющей очень важное значение. Она существенно зависит от уровня образованности и возраста людей.

Частные фирмы предъявляют спрос на фондообразующую продукцию в соответствии с производимыми ими инвестициями и фондообразующими коэффициентами: количествами продукции отраслей производства, необходимыми для

строительств единицы мощности в данной отрасли данной технологии.

Как уже говорилось, государство в ЭДЭМ собирает налоги с частных фирм и домашних хозяйств и тратит эти средства на выплату пособий по безработице, пенсионное обеспечение, оборону. Если выпуск продукции в данной стране в данной отрасли больше, чем затраты этой продукции, вычисление которых описано выше, то страна экспортирует продукцию данной отрасли на глобальный рынок. В противном случае она импортирует эту продукцию. Цены на продукцию отраслей на мировых рынках определяются в соответствии с вариантом известной модели рынка. Именно выпуски отраслей производства, работающих на мировой рынок, пополняют запасы соответствующей продукции на этом рынке.

Спрос на продукцию отраслей предъявляют экономические агенты: правительства стран — в соответствии с принятыми решениями о распределении своих расходов, частные фирмы — в соответствии с решениями о своих инвестициях, домашние хозяйства — в соответствии со структурой потребления.

Цена единицы продукции отрасли зависит от знака и абсолютной величины скорости изменения запаса продукции этой отрасли на рынке. Аналогично описываются рынки труда. Миграцию населения (рабочей силы) из страны в страну предполагается вычислять в демографическом блоке с учетом миграционных законов и мер, предпринимаемых странами по защите своих рынков труда.

В реализованных вариантах модели охарактеризованная общая схема не распространяется на научную, образовательную, культурную и экологическую отрасли.

В реализованных вариантах ЭДЭМ рынок продукции научной отрасли отсутствует: считается, что продукция научной отрасли в стране пропорциональна количеству ученых в ней.

В реализованных вариантах ЭДЭМ рынки услуг культурных отраслей не описаны, функционирование этих отраслей описывается на очень упрощенном уровне - как некоторые расходы домашних хозяйств. Что касается образовательной отрасли, то в реализованных вариантах модели выпуск образовательной отрасли считается пропорциональным государственным расходам и расходам домашних хозяйств на образование.

Государственные расходы на образование являются в реализованных вариантах ЭДЭМ внешними характеристиками, т. е. способ их конкретизации фиксируется в рамках сценариев имитационных экспериментов. Расходы домашних хозяйств на образование определяются в ЭДЭМ структурой потребления.

Модель процесса загрязнения окружающей среды

Выше уже указывалось, что каждая технология характеризуется выбросами загрязнителей в окружающую среду. В модели учитывается два вида загрязнителей первый носит глобальный характер, второй влияет на жизнь людей только в пределах данной страны (региона). Первый далее будет называться глобальным загрязнителем, второй — локальным. При расчете выпуска продукции рентабельными мощностями, обеспеченными рабочей силой (см. выше), определяются также суммарные выбросы загрязнителей обоих типов. По сум-

марным выбросам загрязнителей определяется качество окружающей среды в каждой стране.

Считается, что окружающая среда обладает свойствами самоочищения: с некоторой скоростью загрязнители уничтожаются. В то же время имеется предельная скорость такой переработки, и если загрязнители поступают в среду с большей скоростью, то они начинают накапливаться.

Имеется предельная емкость среды: при превышении некоторого уровня загрязнения механизм самоочищения разрушается и качество среды резко ухудшается — наступает то, что называют экологической катастрофой. От качества среды зависят коэффициенты рождаемости и смертности в демографической модели.

2. Проект распределенной системы прогнозирования демографических процессов в системе регионов

Модель демографического процесса нижнего уровня

Опишем сначала модель эволюции возрастной и половой структуры населения в некотором регионе, которую будет именоваться моделью нижнего уровня или, для краткости, «исходной». Внутренними характеристиками в этой модели будут численности $x_{f,t,a}$, $x_{m,t,a}$, тех женщин и мужчин, которые в году t при исполнении им a лет были живы. Здесь $t = t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$. ($x_{f,t,0}$, $x_{m,t,0}$, $t = t_0 + 1, \dots, T$ — это количество девочек и мальчиков, родившихся в году t .) Любому человеку, которому в году t исполняется a лет, в следующем $t + 1$ - году должно исполниться $a + 1$ лет. Поэтому если бы люди не умирали, то имели бы место следующие равенства $x_{m,t+1,a+1} = x_{m,t,a}$, $x_{f,t+1,a+1} = x_{f,t,a}$.

В реальности, однако только часть людей доживает до своего следующего дня рождения. Обозначим через $\beta_{m,t,a}$ долю мужчин, среди тех, которые в момент исполнения им a лет в году t были живы, но которые не дожили до своего следующего дня рождения. По определению имеем

$$\beta_{m,t,a} = \frac{x_{m,t,a} - x_{m,t+1,a+1}}{x_{m,t,a}},$$
$$t = t_0, t_0 + 1, \dots, T - 1, a = 0, 1, \dots, 99. \quad (2.1)$$

Примем пока гипотезу о том, что характеристика $\beta_{m,t,a}$, $t = t_0, t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$, понимаемая как функция целочисленных аргументов t и a , которую в демографии принято называть силой смертности, является внешней характеристикой в составляемой модели, т. е. она не зависит от значений $x_{f,t,a}$, $x_{m,t,a}$, $t = t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$. Характеристика $\beta_{m,t,a}$, конечно, существенно зависит от возраста a . Зависимость этой характеристики от номера года t может определяться многими социальными, политическими, экологическими факторами: уровень жизни, уровень медицинского обслуживания, уровень загрязнения окружающей среды, войны, изменения социально-экономической формации, эпидемии и т.д. Через $\beta_{f,t,a}$ обозначается характеристика, аналогичная характеристике $\beta_{m,t,a}$, но относящаяся к женщинам, которая также считается внешней характеристикой составляемой модели. Тогда

$$\beta_{f,t,a} = \frac{x_{f,t,a} - x_{f,t+1,a+1}}{x_{f,t,a}}. \quad (2.2)$$

Перепишем эти соотношения в более удобном виде

$$x_{m,t+1,a+1} = x_{m,t,a} (1 - \beta_{m,t,a}), \quad (2.3)$$

$$x_{f,t+1,a+1} = x_{f,t,a} (1 - \beta_{f,t,a}), \quad (2.4)$$

$$t = t_0, t_0 + 1, \dots, T - 1, \quad a = 0, 1, \dots, 99.$$

В этих уравнениях не учитывается миграция населения в регион и за его пределы. Через $y_{m,t,a}$, $y_{f,t,a}$, $t = t_0, t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$, обозначается количество въехавших в рассматриваемый регион на ПМЖ соответственно мужчин и женщин в году t имеющими в момент выезда

(въезда) a лет. Через $z_{m,t,a}$, $z_{f,t,a}$ $t = t_0, t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$, обозначается количество выехавших из рассматриваемого региона, соответственно, мужчин и женщин в году t , имеющих в момент выезда (въезда) a лет. Соотношения (2.3), (2.4) примут вид

$$x_{m,t+1,a+1} = x_{m,t,a} (1 - \beta_{m,t,a}) + y_{m,t,a} - z_{m,t,a}, \quad (2.5)$$

$$x_{f,t+1,a+1} = x_{f,t,a} (1 - \beta_{f,t,a}) + y_{f,t,a} - z_{f,t,a}, \quad (2.6)$$

$$t = t_0, t_0 + 1, \dots, T - 1, \quad a = 0, 1, \dots, 99.$$

Начальные значения

$$x_{m,t_0,a} = x_{m,0,a}, \quad x_{f,t_0,a} = x_{f,0,a}, \quad a = 0, 1, \dots, 100, \quad (2.7)$$

внутренних характеристик считаются (как и во всех моделях развивающихся во времени процессов) внешними характеристиками составляемой модели. Составленные соотношения (2.5) – (2.7) еще не образуют замкнутой модели, поскольку из этих соотношений нельзя вычислить при известных внешних характеристиках внутренние характеристики $x_{m,t,0}$, $x_{f,t,0}$, $t = t_0 + 1, \dots, T - 1$ — количество мальчиков и девочек, родившихся в году t .

Обозначим через $\gamma_{m,t,a}$ количество мальчиков, родившихся от женщин, имеющих в году t возраст a в расчете на одну женщину. Обозначим через $\gamma_{f,t,a}$ аналогичную величину для девочек. Эти характеристики будут считаться внешними в составляемой модели, т. е. не зависящими от значений $x_{m,t,a}$, $x_{f,t,a}$. В демографии они называются коэффициентами рождаемости. В демографии принято считать, что $\gamma_{m,t,a}$ и $\gamma_{f,t,a}$ равны нулю при $a < 15$ и $a > 50$. Таким образом,

$$x_{m,t,0} = \sum_{a=15}^{a=50} x_{f,t,a} \gamma_{m,t,a}, \quad t = t_0 + 1, t_0 + 2, \dots, T, \quad (2.8)$$

$$x_{f,t,0} = \sum_{a=15}^{a=50} x_{f,t,a} \gamma_{f,t,a}, \quad t = t_0 + 1, t_0 + 2, \dots, T. \quad (2.9)$$

Нетрудно видеть, что соотношения (2.5) - (2.9) являются замкнутой моделью, если считать характеристики $x_{m,t,a}$, $x_{f,t,a}$, $t = t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$, внутренними, характеристики $\beta_{m,t,a}$, $\beta_{f,t,a}$, $\gamma_{m,t,a}$, $\gamma_{f,t,a}$, $t = t_0, t_0 + 1, \dots, T$, $a = 0, 1, \dots, 100$, $y_{m,t_0,a}$, $y_{f,t_0,a}$, $z_{m,t,a}$, $z_{f,t,a}$, $x_{m,0,a}$, $x_{f,0,a}$, $a = 0, 1, \dots, 100$, — внешними. В самом деле, используя соотношения (2.5), (2.6), записанные для $t = t_0$, получим из них по известным $x_{m,t_0,a} = x_{m,0,a}$, $x_{f,t_0,a} = x_{f,0,a}$, $\beta_{m,t_0,a}$, $\beta_{f,t_0,a}$, $y_{m,t_0,a}$, $y_{f,t_0,a}$, $z_{m,t,a}$, $z_{f,t,a}$ $a = 0, 1, \dots, 100$, значения $x_{m,t_0+1,a}$, $x_{f,t_0+1,a}$, $a = 1, \dots, 100$. Далее, используя полученные значения $x_{m,t_0+1,a}$, $x_{f,t_0+1,a}$, $a = 1, \dots, 100$, и известные значения $\gamma_{m,t_0+1,a}$, $\gamma_{f,t_0+1,a}$ $a = 15, 16, \dots, 50$, по формулам (2.8), (2.9) вычислим значения $x_{m,t_0+1,0}$, $x_{f,t_0+1,0}$. Таким образом, определены внутренние характеристики $x_{m,t,a}$, $x_{f,t,a}$ при $t = t_0 + 1$ и всех значениях целочисленной переменной a . Точно таким же способом вычисляются значения внутренних характеристик $x_{m,t,a}$, $x_{f,t,a}$ при $t = t_0 + 2$ и всех значениях целочисленной переменной a , а значит, и при всех остальных значениях переменной t . Итак, установлена замкнутость составленной модели.

Система социально-образовательно- профессиональных характеристик населения регионов

Для того чтобы пользоваться изложенной выше моделью «нижнего уровня», необходимо определить ее внешние характеристики, т. е. выполнить ее идентификацию. Что касается начальных значений внешних характеристик, то они определяются переписью населения. В России перепись населения имела место в 2002 г. и очередная перепись должна состояться в 2010 г. Главные проблемы идентификации модели нижнего уровня для некоторого региона лежат в определении коэффициентов рождаемости и смертности и определении миграции населения.

Коэффициенты рождаемости и смертности, а также миграционные процессы зависят от социальной, образовательной, профессиональной структуры населения, от экономических факторов, от мероприятий в законодательной сфере и действий правящих структур, затрагивающих демографические процессы, от миграционного законодательства.

Проблемой является введение характеристик, определяющих эти факторы, которые могут быть различными для различных регионов. В любом случае коэффициенты рождаемости и смертности и их зависимости от социальной, образовательной, профессиональной структуры населения являются результатом прогноза, который выполняется экспертами – демографами.

При выполнении этих прогнозов возможно использование математических моделей, основанных главным образом на средствах математической статистики. Но в любом случае все эти методы будут «внешними» по отношению к модели демографического процесса нижнего уровня.

Заметим также, что, как демографов, так и различные экономические, административные, политические структуры интересуют различные наборы демографических характеристик.

Предметом особого интереса является прогноз влияния на демографические характеристики функционирования законодательных органов, различных мероприятий в социальной сфере.

Например, административные структуры, которые занимаются организацией процесса образования, интересуют прогноз количества первоклассников в каждом году и прогноз количества выпускников средних школ в каждом году. Эти прогнозы на характерных временах порядка десяти лет зависят главным образом от прогноза коэффициентов рождаемости и прогноза миграции.

Прогноз коэффициентов рождаемости зависит от действий законодательной и исполнительной властей, касающихся охраны материнства и детства. Прогноз миграции зависит от характеристик уровня жизни в регионе, по сравнению с уровнем жизни в других регионах.

Административные структуры, связанные со здравоохранением и пенсионным обеспечением, интересуют прогноз количества пенсионеров в каждом году. И так далее. Поэтому необходимы средства, которые позволили бы демографам выполнить свои прогнозы коэффициентов рождаемости и смертности, а также миграции и влияния на эти коэффициенты и миграцию различных факторов. Если коэффициенты рождаемости и смертности конкретизированы, то можно обратиться к модели нижнего уровня и получить прогноз по годам эволюции возрастной и половой структуры населения на протяжении того времени, на котором конкретизирован прогноз.

После этого необходимо задать систему обработки результатов этой модели и получить интересующие характеристики. Те же возможности необходимо обеспечить различным административным, хозяйственно-экономическим, политическим структурам. Эти структуры должны иметь возможность вы-

брать из существующих прогнозов коэффициентов рождаемости и смертности, миграции а также влияния на эти коэффициенты и миграцию различных факторов, те, которым они доверяют в наибольшей степени.

Решение перечисленных проблем видится в разработке модели демографического процесса, реализуемой в режиме распределенных вычислений. Необходима естественно, компьютерная реализация модели нижнего уровня, доступная через Интернет. Необходимо обеспечить доступ к этой модели как профессиональных демографов, так и всех интересующихся демографическим процессом в силу тех или иных причин. При этом каждый пользователь должен иметь возможность или самостоятельно дать прогноз коэффициентов рождаемости и смертности и миграции или выбрать прогноз из тех, которые открыты для пользования на компьютерах профессиональных демографов, или пользоваться теми, которые предоставляется «по умолчанию».

Пользователь также должен конкретизировать способ обработки результатов моделирования, если среди предоставляемых «по умолчанию» отсутствует тот, который ему нужен.

О понятии «когорта»

Для реализации таких возможностей предлагается ввести понятие «когорта» (cohort). Это понятие имеет содержательный и синтаксический аспекты. С содержательной точки зрения когорта – это люди в данном регионе, которые имеют одинаковый набор некоторой совокупности характеристик, описывающих социальный, образовательный, профессиональный и т.д. статус. Синтаксический аспект когорты зависит от языка, на котором будет выполняться программирование.

Остановимся сначала на содержательном аспекте понятия «когорты». Обязательной характеристикой когорты является возрастной промежуток ($ain - aen$) составляющих ее людей. Далее эта характеристика будет называться возрастной парой когорты. В возрастной паре ain – начальный возраст людей, составляющих когорту, aen – конечный возраст этих людей. Случай $ain = aen$ не исключается. Пока рассматривается случай, когда возрастная пара когорты не зависит от времени, т. е. от текущего года. Далее будут определены когорты, где это ограничение отсутствует. Все вводимые далее характеристики когорт являются функциями времени (т. е. года).

Приведем примеры когорт. Когортой можно считать всех людей, имеющих один и тот же возраст a . В этом случае $ain = aen = a$. Изложенная выше модель демографического процесса нижнего уровня позволяет вычислить численность $x_{m,t,a} + x_{f,t,a}$ всех таких когорт, которых всего 101 ($a = 0, 1, \dots, 100$). Когортой можно считать всех людей. В этом случае $ain = 0, aen = 100$. Численность людей в этой когорте в модели нижнего уровня определяется формулой

$$\sum_{a=0}^{100} (x_{m,t,a} + x_{f,t,a}).$$

Когортой можно считать всех женщин, имеющих возраст a . Таких когорт тоже 101 и модель демографического процесса нижнего уровня позволяет вычислить их численность. То же касается всех мужчин, имеющих возраст a . Обязательными характеристиками когорты являются распределение людей по возрастам внутри когорты, а также общая численность людей в когорте.

Если возрастные характеристики (т. е. возрастная пара, возрастное распределение, общая численность людей) являются единственными характеристиками когорты, то когорта называется чисто возрастной. Примером чисто возрастных ко-

горт являются упоминавшиеся когорты, которые составляют все люди имеющие возраст a . Однако все женщины, имеющие возраст a , не являются чисто возрастной когортой, поскольку кроме возрастного промежутка она имеет еще одну характеристику – пол.

Семейство $(ain_i, aen_i)_{i=0,1,\dots,K}$ возрастных когорт назовем полным, если $ain_0=0$, $ain_{i+1}=aen_i+1$, $i=1,2,\dots,K-1$, $aen_K=100$. Полным семейством когорт является семейство, состоящее из единственного элемента – всех людей.

Для иллюстрации понятия когорты проанализируем структуру `population` языка C++, в программе, реализующей ЭДЭМ. Структура `population` декларирует данные, описывающие социально - демографическое состояние населения, учитываемое в модели демографического процесса ЭДЭМ. Приводимую структуру `population` снабдим комментариями (они выполнены по правилам языка C++), которые демонстрируют связь структуры `population` с введенным выше понятием когорты.

```
typedef struct {  
    double ch[6],  
    /* Массив ch является распределением по возрастам детей от  
    0 до 6. Это распределение является характеристикой чисто  
    возрастной когорты с возрастной парой (0,6). Естественно на-  
    звать эту когорту «дошкольники». Знание возрастного распре-  
    деления когорты «дошкольники» позволяет вычислить чис-  
    ленность этой когорты. Когорта «дошкольники» разбивается  
    на две подкогорты – подкогорта «девочки» и подкогорта  
    «мальчики» с той же возрастной парой. В демографической  
    модели ЭДЭМ численность этих подкогорт можно вычислить с  
    помощью массива double ratio[0:100], который декларируется
```

как постоянный массив в декларативной части программы (ecdm.h). Значение $\text{ratio}[a]$ является отношением количества мужчин, имеющих возраст a , к количеству женщин с таким же возрастом. Это отношение в ЭДЭМ вычисляется по начальным данным, которые содержат возрастные распределения как женщин, так и мужчин в начальном году. Аналогичным образом дело обстоит с теми вводимыми когортами, которые не делятся по половому признаку. Однако, для всякой когорты это деление можно выполнить с помощью массива `double ratio[0:100].*/`

`sc[11], /*` Массив `sc` является распределением по возрастам детей от 7 до 17. Это распределение является характеристикой чисто возрастной когорты с возрастной парой (7,17). Естественно назвать эту когорту «школьники». Другим названием этой когорты может быть «люди с незаконченным средним образованием». Знание возрастного распределения когорты «школьники» позволяет вычислить численность этой когорты. `*/`

`st[5], /*` Массив `st` является распределением по возрастам юношей от 18 до 22, которые обучаются в ВУЗах. Это распределение является характеристикой когорты (не являющейся чисто возрастной) с возрастной характеристикой (7,17). Естественно назвать эту когорту «студенты». Другим названием этой когорты может быть «люди с незаконченным высшим образованием». Знание возрастного распределения когорты «студенты» позволяет вычислить численность этой когорты. `*/`

`lb[42], /*` Массив `lb` является распределением по возрастам людей от 18 до 59. Это распределение является характеристикой когорты с возрастной характеристикой (18,59). Характеристикой данной когорты является также уровень образованности составляющих ее людей. Именно, данную когорту составляют люди, имеющие аттестат зрелости, т. е. свидетельство об

окончании среднего учебного заведения, но не имеющие высшего образования. */

as[3], /* Массив as является распределением по возрастам людей от 23 до 25. Это распределение является характеристикой когорты с возрастной парой (23,25). Данную когорту составляют люди, окончившие ВУЗ и обучающиеся в аспирантуре (т. е. имеющие незаконченное поствысшее образование). */

ow[37], /* Массив ow является распределением по возрастам людей от 23 до 59. Это распределение является характеристикой когорты с возрастной парой (18,59). Характеристикой данной когорты является также уровень образованности составляющих ее людей. Именно, данную когорту составляют люди, имеющие высшее образование. */

sci[34], /* Массив sci является распределением по возрастам людей от 26 до 59. Это распределение является характеристикой когорты с возрастной парой (26,59). Характеристикой данной когорты является также уровень образованности составляющих ее людей. Именно, данную когорту составляют люди, окончившие аспирантуру, т. е. имеющие поствысшее образование. */

pens[41], /* Массив pens является распределением по возрастам людей от 60 до 100. Это распределение является характеристикой когорты с возрастной парой (60,100). В ЭДЭМ люди с такими возрастными считаются пенсионерами.*/

reople[100], /* Массив reople является распределением по возрастам людей от 0 до 100. Это распределение является характеристикой когорты чисто возрастной когорты с возрастной парой (0,100). */

men[100], /* Массив men является распределением мужчин по возрастам от 0 до 100. Это распределение является характеристикой когорты с возрастной парой (0,100). */

women[100], /* Массив women является распределением женщин по возрастам от 0 до 100. Это распределение является характеристикой когорты с возрастной парой (0,100). */

conslb[42][5], /* Массив conslb является характеристикой когорты, обозначенной выше lb. Значение conslb[a][p] есть доля зарплаты, которую получает человек возраста a со средним образованием на рынке труда и которая (доля) тратится им на приобретение для конечного потребления продукта p-й отрасли. */

consow[37][5], /* Массив consow является характеристикой когорты, обозначенной выше ow. Значение consow[a][p] есть доля зарплаты, которую получает человек возраста a с высшим образованием на рынке труда и которая (доля) тратится им на приобретение для конечного потребления продукта p-й отрасли. */

conssci[34][5], /* Массив conssci является характеристикой когорты, обозначенной выше sci. Значение conssci[a][p] есть доля зарплаты, которую получает человек возраста a со средним образованием на рынке труда и которая (доля) тратится им на приобретение для конечного потребления продукта p-й отрасли. */

conspens[41][5], /* Массив conspens является характеристикой когорты, обозначенной выше pens. Значение conspens [a][p] есть доля пенсии, которую получает человек возраста a и которая тратится им на приобретение для конечного потребления продукта p-й отрасли. */

partow[83], /*Массив partow является характеристикой когорты, имеющей возрастную пару (18,100). Значение partow[a] есть доля людей возраста a+18, имеющих или незаконченное или законченное высшее образование.*/

partsci[78], /*Массив partsci является характеристикой когорты, имеющей возрастную пару (18,100). Значение partsci[a] есть

доля людей возраста $a+23$, имеющих или незаконченное или законченное поствысшее образование.*/
} population;

Итак, модель демографического процесса в ЭДЭМ оперирует когортами и их характеристиками. Поэтому можно попытаться описать демографический процесс, оперируя только понятием «когорта».

Как уже отмечалось, синтаксический аспект понятия «когорта» зависит от используемого языка программирования. Далее предполагается, что программирование будет выполняться на языке C++. Переход на другой язык программирования потребует некоторых технических усилий. При использовании средств языка C++ наиболее естественным средством для описания когорты является class.

Идея состоит во введении «стандартного» класса «когорты». Пользователь должен конкретизировать экземпляры этого класса так, чтобы совокупность этих экземпляров определяла замкнутую модель демографического процесса. Для этого необходимо, чтобы определяемые пользователем когорты содержали семейство когорт, являющиеся полным в том смысле, о котором говорилось выше, как для женщин, так и для мужчин, а также информацию и процедуры, позволяющие рассчитать коэффициенты рождаемости и смертности с учетом вводимой пользователем структуры когорт.

Кроме полных семейств когорт для мужчин и женщин, которые дают возможность обратиться к модели демографического процесса нижнего уровня, в определяемой структуре когорт могут присутствовать различные когорты, которые являются элементами обработки результатов моделирования. Такие когорты снабжаются соответствующим признаком, означающим, что они предназначены для обработки инфор-

мации уже после обращения к модели демографического процесса нижнего уровня.

Итак, пользователь формирует экземпляры когорт так, чтобы сформированное семейство когорт было полным и содержало информацию и процедуры, делающие замкнутой модель демографического процесса нижнего уровня. Кроме того, пользователь формирует экземпляры когорт, которые предназначены для обработки результатов расчетов по модели нижнего уровня. Система моделирования демографического процесса выполняет расчеты по модели нижнего уровня. Пользуясь полученными половыми и возрастными распределениями, можно выполнить расчет характеристик всех экземпляров когорт, сформированных пользователем.

С помощью режима распределенных вычислений пользователь должен иметь возможность или самостоятельно сформировать структуру когорт, содержащих всю информацию и процедуры для расчета демографического процесса, или выбрать ту структуру когорт из имеющихся на различных компьютерах в открытом доступе, которую он считает наиболее приемлемой для расчета интересующего его демографического процесса. Он должен иметь возможность также сформировать когорты, характеристики которых ему интересны, и получить эти характеристики на свой компьютер после конкретизации всего, что необходимо для обращения к модели демографического процесса нижнего уровня.

```
class cohort
{
  public:
```

```
cohort();/*процедура – конструктор, с помощью которой  
инициализируется начальное распределение людей по воз-  
растам */
```

```
int ain; /*начальный возраст людей когорты*/
```

```
int aen; /*конечный возраст людей когорты*/
```

```
double partdistr[aen-ain+1]; /* доля людей в данной когор-  
те с данным возрастом в общем количестве людей с данным  
возрастом*/
```

```
Boolean sex; /* пол людей когорты*/
```

```
char nationality; /* национальность людей когорты*/
```

```
double agedistr[aen-ain+1]; /*возрастное распределение  
людей в когорте*/
```

```
double amount; /* общее количество людей когорты*/
```

```
double partsecondaryedu[aen-ain+1] /*доля людей когорты  
со средним образованием*/
```

```
double parthigheredu[aen-ain+1] /*доля людей когорты с  
высшим образованием*/
```

```
double partpostedu[aen-ain+1] /*доля людей когорты с по-  
ствысшим образованием*/
```

```
double consum[AmountOfConsumptionKinds] /*доля  
потребления товаров различных видов*/
```

```
void Initialdistr();/*процедура, определяющая распределе-  
ние по возрастам людей в когорте в начальном году*/
```

```
void Inputcohort();/*процедура, определяющая количество  
людей в когорте с возрастом ain в следующем году*/
```

```
void Time();/*процедура, определяющая распределение  
людей по возрастам в когорте в следующем году*/
```

```
}
```

В качестве примера опишем класс `people`, являющийся когортой всех людей в рассматриваемом регионе.

```
class people
{ public:
    people(); /*процедура - конструктор, конкретизирующая
    начальное распределение людей в когорте*/
    int ain=0;
    int aen=100; /*начальный и конечный возраст людей когор-
    ты people*/
    double partdistr[aen-ain+1]; /* доля людей в данной когорте
    с данным возрастом в общем количестве людей данного воз-
    раста; все числа этого массива равны 1 */
    enum sex=0; /* пол людей когорты; перечислимый тип, 0 –
    люди в когорте не различаются по полу,
    men – мужчины, women – женщины*/
    enum nationality; /* национальность людей когорты*/
    double agedistr[aen-ain+1]; /*возрастное распределение
    людей в когорте*/
    double amount; /* общее количество людей когорты*/
    double partsecondaryedu[aen-ain+1] /*доля людей когорты
    со средним образованием*/
    double parthigheredu[aen-ain+1] /*доля людей когорты с
    высшим образованием*/
    double partpostedu[aen-ain+1] /*доля людей когорты с по-
    ствысшим образованием*/
    double consum[AmountOfConsumptionKinds] /*доля
    потребления товаров различных видов*/
    void Initialdistr();/*процедура, определяющая распределе-
    ние по возрастам людей в когорте в начальном году*/
    void Inputcohort();/*процедура, определяющая количество
    людей в когорте с возрастом ain в следующем году*/
```

```
void Time();/*процедура, определяющая распределение  
людей по возрастам в когорте в следующем году*/  
}
```

Таким образом, в общих чертах описан проект распределенной системы прогнозирования демографических процессов в системе регионов. Перед тем, как перейти к описанию компьютерной реализации макета эколого-демографо-экономической модели в режиме распределенных вычислений, которая дается в разд. 4, обсудим общую концепцию построения инструментальной системы распределенного моделирования.

3. Концепция построения инструментальной системы распределенного моделирования

В работах [16-19] был предложен подробный объектно-ориентированный анализ сложных моделей, позволяющий декомпозировать сложную систему на достаточно простые элементы, а затем осуществлять из этих элементов полноценный синтез модели. В данной работе представляется более уместным для описания анализа и синтеза модели отдать предпочтение языку динамических систем перед языком объектного анализа. Это предпочтение вовсе не означает, что идеи и методы работ [1,2] устарели или что объектно-ориентированный анализ непригоден для рассматриваемой проблематики.

Во-первых, хотелось бы уметь интегрировать в разрабатываемую инструментальную среду реально существующие модели. Залогом успеха здесь может быть минимальность дополнительных требований к уже существующей модели со стороны инструментальной среды. С точки зрения этой минимальности представление произвольной модели динамическим процессом представляется весьма универсальным и в то же время достаточно необременительным.

Во-вторых, с точки зрения автора, системно-динамический и объектно-ориентированный языки описания сложных систем являются функционально эквивалентными и взаимодополняющими (как, например, волновое и корпускулярное описание природы света). Поэтому описание такого явления, как моделирование сложных систем, на двойствен-

ном языке, возможно, выявит его некоторые неотмеченные ранее грани.

Компонента - основная составляющая распределенной модели

Основными понятиями данной работы будут понятия комплекса и компоненты. Это связанные между собой понятия, способные переходить друг в друга. Так, комплекс, внутренняя структура которого сложна и образована множеством компонент, в то же время извне может восприниматься как единая компонента. В то же время, у любой из компонент подробное исследование может выявить внутреннюю составную структуру, образующую комплекс.

Попробуем в ближайших двух разделах определить, что же такое компонента модели. Основной абстракцией, которой мы будем оперировать далее, будет динамическая система

$$\vec{x}_{i+1} = \vec{F}(\vec{x}_i, \vec{a}_i, \Delta t), \quad (3.1)$$

где вектор \vec{x}_i - набор внутренних или фазовых характеристик системы, вектор \vec{a}_i - набор ее внешних характеристик или параметров, Δt - интервал модельного времени между двумя смежными шагами динамического процесса, i и $i+1$, \vec{F} - некий алгоритм, детерминированный или стохастический, позволяющий для любого i по значениям \vec{x}_i , \vec{a}_i и Δt однозначно определять реализацию фазовых характеристик \vec{x}_{i+1} на следующем шаге динамического процесса. Существование такого алгоритма означает замкнутость системы, т. е. опреде-

ленную ее независимость от внешнего мира, а именно: задавая \vec{x}_0 и \vec{a}_i , можно рассчитать внутренние характеристики системы в любой момент модельного времени. Здесь \vec{x}_0 - начальное состояние системы, вектор внешних характеристик \vec{a}_i на самом деле осуществляет связь нашей системы с внешним миром. Динамическую систему вида (3.1) будем называть моделью в тех случаях, когда нужно подчеркнуть ее вхождение в качестве компоненты в более крупную модель, – подмоделью или компонентой, а в тех случаях, когда нужно подчеркнуть ее собственную многокомпонентную внутреннюю структуру – метамоделью или комплексом.

Реакция компоненты на события

Известно, что реальные сложные системы, являющиеся предметом моделирования, в зависимости от определенных сочетаний своих характеристик и характеристик окружающего мира, способны порой достаточно резко менять свое поведение. Для отражения этого их свойства введем формализованное определение события. Будем считать, что события могут происходить в самом начале интервала модельного времени Δt , при переходе системы от i -го к $i+1$ -му состоянию. С моделью может быть связано несколько событий, событие $k, k=1, \dots, K$ состоит в том, что некая скалярная функция $\Phi_k(\vec{x}, \vec{a})$ обращается в ноль на наборе характеристик модели \vec{x}_i, \vec{a}_i в начале очередного интервала моделирования: $\Phi_k(\vec{x}_i, \vec{a}_i) = 0$. Далее, можно считать, что имеется некоторое количество способов поведения модели, задающееся различными алгоритмами $\vec{F}_m(\vec{x}, \vec{a}, \Delta t), m=1, \dots, M$, и однозначное отображение множества 2^K всевозможных реализаций всех

связанных с моделью событий, во множество M различных алгоритмов ее поведения, которое и задает реакцию модели на произошедшее событие.

Заметим, что, стартовав с системно-динамического описания компоненты модели, мы пришли к двойственному, объектно-ориентированному описанию: компонента, определенная выше, задает класс объектов со свойствами \vec{x}_i, \vec{a}_i , методами $\vec{F}_m(\vec{x}, \vec{a}, \Delta t), m = 1, \dots, M$, переключения между методами задаются событиями $\Phi_k(\vec{x}_i, \vec{a}_i) = 0, k = 1, \dots, K$.

Анализ метамодели (комплекса)

Предположим, что для некоторого явления мы исследуем модель, которая описывается динамической системой (3.1). В работе [1] было показано, что если алгоритм динамического процесса задается функционально, то при некоторых не слишком обременительных условиях, накладываемых на функцию \vec{F} , систему (3.1) с любой наперед заданной точностью можно заменить несколькими динамическими системами, просто разбивая вектор \vec{x}_i на несколько подвекторов, и синхронизируя подсистемы между собою в точках событий. Далее, попытаемся проиллюстрировать еще один источник возникновения нескольких динамических систем, при углубленном изучении модели, первоначально описывавшейся одной системой.

Мы предполагаем замкнутость нашей модели, т. е. некоторую ее изолированность от внешнего мира, а именно: зная векторы \vec{x}_0 и \vec{a} , мы можем рассчитать траекторию нашей

системы на любом шаге. При этом о внешнем мире нам вроде бы ничего и не надо знать. На самом же деле все в этом мире связано, и связь внешнего мира с нашей моделью осуществляется именно через внешние переменные \vec{a} модели, а также через вид функции \vec{F} , который как «вещь в себе» никогда нам не известен, а всегда придуман исследователем и является синтезом его знаний о данном явлении и о его связях с внешним миром, на настоящем этапе исследования. Обычно функция \vec{F} придумывается исследователем как некоторая комбинация «элементарных» (т. е. хорошо ему известных, ставших стандартными **элементами** его языка) функций, при этом компоненты вектора \vec{a} есть различные коэффициенты в этой комбинации. Или же функция \vec{F} может реализовываться некоторым придуманным исследователем компьютерным алгоритмом, умеющим превращать векторы \vec{x}_i и \vec{a} в вектор \vec{x}_{i+1} . На начальных этапах моделирования компоненты \vec{a} можно считать постоянными или же меняющимися во времени по некоторому эмпирическому закону $\vec{a}(t)$. Однако при более глубоком изучении явления такой подход может перестать устраивать исследователя, могут, например, возникнуть некоторые соображения о природе закона изменения $\vec{a}(t)$ и, как следствие может возникнуть желание описать изменения всех компонент вектора \vec{a} или же какой-либо их части также динамической системой $\vec{a}_{i+1} = \vec{\Psi}(\vec{a}_i, \vec{b}_i, \Delta t)$. В этой динамической системе \vec{a} - уже внутренние переменные, а внешними являются компоненты вектора \vec{b} . Если еще при этом окажется, что вектор \vec{b} , это на самом деле по-другому названный вектор \vec{x} , а над моделью $\vec{a}_{i+1} = \vec{\Psi}(\vec{a}_i, \vec{b}_i, \Delta t)$ уже давно и успешно работает некий другой исследователь, то можно объединить усилия этих исследова-

телей и построить к их общей пользе объединяющую две модели метамодель: $\bar{x}_{i+1} = \bar{F}(\bar{x}_i, \bar{a}_i, \Delta t)$, $\bar{a}_{i+1} = \bar{\Psi}(\bar{a}_i, \bar{b}_i, \Delta t)$.

Конечно, в реальной жизни такого полного и быстрого замыкания двух моделей, скорее всего, не получится. Вероятнее будет задействовано большее количество подмоделей и меньшее количество параметров.

Синтез метамодели (комплекса)

Предположим, наша модель описывается несколькими динамическими системами, как это было описано в предыдущем пункте. Возникает вопрос, как построить из них работающую метамодель.

Во-первых, необходим информационный синтез подмоделей, т. е. необходимо указать для каждой подмодели, какие ее внутренние переменные являются внешними переменными каких-либо, и каких именно подмоделей. И наоборот, какие из ее внешних переменных на самом деле явно моделируются какой-то (и какой именно) подмоделью. Будем считать, что мы располагаем N компонентами A_k , описываемыми динамическими системами $\bar{x}_{i+1}^k = \bar{F}^k(\bar{x}_i^k, \bar{a}_i^k, \Delta t)$, $k = 1, \dots, N$. На практике вполне может оказаться, что результаты работы некоторых компонент (т. е. их внутренние переменные) определяют внешние переменные некоторых других компонент не непосредственно, а через определенные функциональные или алгоритмические зависимости, подобно тому, как это бывает в задаче наблюдения: мы не умеем непосредственно измерять интересующие нас величины, но умеем измерять некоторые другие, функционально связанные с пер-

выми, и на основе измерений и зависимостей вычислять их. Однако мы имеем право считать, что все подобные зависимости представлены в универсальной форме (3.1), т. е. в форме компонент, и таким образом входят в число компонент нашего комплекса. Далее, для каждой из $N(N-1)$ пар $(A_i, A_j), i \neq j$, можно построить матрицу коммутации $Q_{i,j}$ размера $n_i \times m_j$, где n_i - размерность вектора \vec{a}^i компоненты A_i , а m_j - размерность вектора \vec{x}^j компоненты A_j . На пересечении p -й строки ($1 \leq p \leq n_i$) и q -го столбца ($1 \leq q \leq m_j$) этой матрицы стоит 1, если внешняя переменная a_p^i компоненты A_i явно моделируется внутренней переменной x_q^j компоненты A_j ; и 0 - в противном случае. Задание матриц коммутации $Q_{i,j}, 1 \leq i, j \leq N; i \neq j$ полностью решает вопрос информационного синтеза комплекса из компонент. Здесь следует заметить, что при фиксированном i , и при $1 \leq j \leq N; i \neq j$, наличие единицы в фиксированной строке более чем в одной из матриц $Q_{i,j}$, так же как и наличие в одной строке более одной единицы, свидетельствовало бы о моделировании одной величины несколькими способами и, следовательно, ставило бы вопрос о согласованности таких моделей. Стало быть, подобная ситуация должна вызывать если и не сообщение о фатальной ошибке при коммутации компонент комплекса, то по крайней мере очень серьезное предупреждение с призывом тщательно проверить согласованность компонент. С другой стороны, наличие нескольких единиц в столбце матрицы $Q_{i,j}$, а также при фиксированном j и при $1 \leq i \leq N; i \neq j$, наличие единицы в фиксированном столбце более чем одной из матриц $Q_{i,j}$, говорит о том, что внутренняя переменная одной из компо-

нент используется в качестве внешней переменной более чем одной компоненты, что не противоречит нашей концепции, и должно быть разрешено. Следует также заметить, что поскольку, как следует из сказанного выше, единица – достаточно «редкий гость» в матрицах коммутации, то при описании комплекса на формальном языке, будет проще специальным оператором языка описывать только существующие коммутации вместо выписывания всех матриц коммутации, которые должны автоматически строиться и проверяться на этапе компиляции описания комплекса.

Во-вторых, возникает вопрос о синхронизации во времени работы подмоделей. Предлагается следующий механизм синхронизации модельного времени компонент комплекса.

- Из соображений выбора масштаба и точности моделирования выбирается некий стандартный интервал моделирования Δt .
- На интервале моделирования Δt аппроксимируются значения векторов $\vec{x}(t), \vec{a}(t)$. Это может быть линейная аппроксимация по двум точкам (\vec{x}_i, \vec{a}_i) и $(\vec{x}_{i+1}, \vec{a}_{i+1})$ на концах этого интервала или аппроксимация по уже существующим точкам моделирования от начала динамического процесса, или же вычисление одной или нескольких промежуточных точек на интервале Δt (т. е. фактически, счет с меньшим шагом моделирования), и затем аппроксимация одним из известных методов, по этим точкам. Выбор того или иного способа аппроксимации зависит от масштаба моделирования и желаемой точности модели.

- Далее, для каждого события каждой подмодели ищется решение t уравнения $\Phi_k(\vec{x}(t), \vec{a}(t)) = 0$, $k = 1, \dots, K$ на интервале Δt . Если таких решений нет, в качестве следующего интервала моделирования снова выбирается стандартный интервал моделирования Δt , если же решения есть, то наименьшее из них будет правым концом следующего интервала моделирования. Таким образом, интервал моделирования может уменьшаться, с тем, чтобы ни одна из подмоделей не «проскочила» очередное событие, которое может существенно повлиять на ее поведение.
- Необходимо избегать «дурной бесконечности» событий, руководствуясь принципом достаточности конечного числа событий на любом конечном интервале времени, сформулированном в [2], т. е. точек накопления событий на конечном интервале времени не должно быть.

В заключение следует отметить, что комплекс, внутри состоящий из многих компонент, вовне может проявляться в качестве единой компоненты. Введем следующую операцию объединения компонент комплекса.

1. Внутренними переменными комплекса объявляется объединение внутренних переменных всех его компонент.
2. Методами комплекса объявляется объединение всех методов его компонент.
3. Событиями комплекса объявляется объединение всех событий его компонент.
4. Внешними переменными комплекса объявляется объединение всех внешних переменных его компонент, из

которого исключаются все те переменные a_p^i , для которых p -я строка одной из матриц коммутации $Q_{i,j}$, $1 \leq i, j \leq N; i \neq j$ содержит единицу.

Мы видим, что тогда как изнутри комплекс имеет сложный многокомпонентный состав, извне он, в соответствии с данным в начале работы определением, вполне может быть воспринят как единая компонента. Единственное отличие от данного выше определения компоненты – возможность параллельного (т. е. одновременного) выполнения нескольких методов. Это отличие можно снять, например, разрешив компоненте одновременно выполнять несколько потоков (процессов), состоящих в чередовании некоторого числа выполняемых последовательно методов, как это было в MISS [16]. Так или иначе, определяющим остается то, что задавая для комплекса, рассматриваемого как единая компонента, начальные данные и значения внешних переменных, можно однозначно определить его внутренние переменные на любом шаге моделирования.

Таким образом, любая модель может быть представлена, с одной стороны, агрегированно, как единая компонента, а с другой стороны, подробно, причем с произвольной степенью подробности, как комплекс компонент, в котором каждая из компонент тоже в свою очередь допускает представление в виде комплекса, тем самым позволяя реализовывать концепцию мультимоделирования, т. е. построения семейства моделей разной степени подробности, для изучения одного и того же явления.

Общая архитектура инструментальной системы распределенного имитационного моделирования

Система распределенного имитационного моделирования представляется пиринговой сетью средней степени централизации наподобие популярной сегодня в определенных кругах пользователей Интернета файлообменной сети e-Donkey.

Несколько десятков серверов глобальной сети имитационного моделирования должны выполнять следующие функции обслуживания нескольких тысяч рабочих станций.

- Регистрация клиентских рабочих станций и предоставляемых ими в сеть разделяемых ресурсов, т. е. моделей, компонент моделей, данных, алгоритмов и т.д.
- Хранение и постоянное поддержание баз данных модельных ресурсов в сети, т. е. какими рабочими станциями предоставлены в сеть те или иные модели, компоненты моделей и прочие ресурсы. Какие из рабочих станций реально присутствует в сети, кто из них и когда был доступен в последний раз, возможно, расписаний присутствия рабочих станций в сети.
- Поиск ресурсов в сети, т. е. уже существующих моделей, компонент моделей, необходимых данных и алгоритмов (примерно так, как осуществляется поиск различных файлов в современных пиринговых файлообменных сетях).
- Осуществление функций посредников стандартных запросов рабочих станций.

- Осуществление некоторых организационных функций, например, возможности нескольким рабочим станциям договориться о проведении совместной сессии имитационных экспериментов определенной продолжительности, с началом в определенное время, или же спланировать время проведения такой сессии в соответствии с их расписаниями присутствия в сети.

Клиентское программное обеспечение рабочих станций такой сети вполне могло бы базироваться на описанных выше принципах анализа и синтеза модели, а именно: основу программного обеспечения рабочей станции составляет «система программирования» на некотором непроцедурном языке описания моделей и их компонент. Эта система программирования должна включать редактор (в том числе желательно и с графическим интерфейсом) описания моделей и их компонент. При описании компонент модели той или иной конструкции должна быть возможность ссылки на найденные уже существующие в сети компоненты, подходящие на роль компонент описываемой конструкции (наподобие e2k:-ссылок сети e-Donkey). Далее, система программирования клиентской части должна включать средства отладки описаний отдельных компонент и всей модели в целом. Далее, в клиентской части должен присутствовать сборщик модели, создающий базу характеристик компонент модели, проверяя при этом ее целостность.

Блок клиентской части инструментальной системы, ответственный за организацию имитационных экспериментов, должен с помощью существующих интеграционных технологий вызывать как локальные, так и удаленные компоненты модели по их известным адресам, передавая им необходимые

характеристики из базы данных модели и принимая от них обратно в базу обработанные характеристики. Служебные методы, предоставляемые системой, также должны быть рассчитаны как на локальное, так и на удаленное использование. На этапе имитационного эксперимента рабочая станция, проводящая эксперимент, работает напрямую, минуя сервера, с рабочими станциями, содержащими используемые в модели компоненты. Как и в современных пиринговых сетях, серверы здесь нужны лишь на этапе поиска партнеров.

Уже отлаженные, работающие модели и их компоненты могут быть объявлены доступными для всеобщего использования и пополнить фонд доступных в сети моделей и их компонент. При этом автоматически становятся доступными также и описания модели и ее компонент на языке описаний, по которым можно судить о ее адекватности тем или иным способам применения в других моделях. Таким образом, каждая рабочая станция может предоставлять в общее пользование как свой вычислительный потенциал, так и существующие наработки в области моделирования.

Предложенная концепция могла бы применяться и более широко как универсальное средство совместного использования в сети разнородных и разноплатформенных информационных и алгоритмических ресурсов, что могло бы существенно расширить круг ее потенциального применения. Тем самым у инструментальных систем имитационного моделирования появился бы шанс превратиться из экзотических программных продуктов, предназначенных для узкого круга специалистов в области имитации, в массовое и универсальное средство объединения ресурсов Интернета для совместного выполнения широкого круга задач.

Предложенная концепция построения глобальной пиринговой сети имитационного моделирования может быть реализована на базе технологии IARnet, разработанной в Институте системного анализа РАН [18].

Как описываются комплексы и компоненты

Для описания комплексов и компонент необходим специализированный непроцедурный язык описаний классов комплексов и компонент. Одним из первых языков подобного рода был предложенный в [16] язык описания спецификаций MISS, затем в разное время обретали и теряли популярность такие языки, как IDL, UML, различные языки описания спецификаций на основе XML, наконец, Slice. В настоящее время в этой области наблюдается определенный разброд и шатание. Поэтому, в данной работе мы не остановимся на каком-либо конкретном из перечисленных выше языков, тем более, что их возможности а стало быть, и сложность их реализации, существенно превосходят функциональные потребности данного проекта, а сконцентрируем внимание именно на этих функциональных потребностях.

Язык описания комплексов и компонент

В данном разделе разобран специализированный непроцедурный язык, на котором должны составляться спецификации классов компонент модели. Отметим, что хотя аккуратнее употреблять термин "спецификация класса компонент", мы ради краткости будем пользоваться словосочетанием "спецификация компоненты". Лингвистические формулы записываются ниже в стандартной нотации, согласно которой

служебные слова выделяются жирным шрифтом, разделители - двойными кавычками, а необязательные включения - квадратными или фигурными скобками, причем фигурные скобки указывают на возможность повторения. Разбор языка начнем с двух нетривиальных конструкций, применяемых в спецификациях всех типов. Это - определение структуры данных и оператор коммутации.

Определение типов записей фазовых переменных и констант

Собирая модель, инструментальная система автоматически генерирует ее базу данных, руководствуясь при этом содержанием специальных параграфов спецификаций компонент. Речь идет о параграфах определений типов фазовых переменных и констант. Их синтаксис различается только открывающей ключевой конструкцией, а в остальном сводится к правилам определения структур данных. Формально определение структуры данных есть набор определений списков полей, т.е. набор списков идентификаторов, завершающихся указаниями типов поименованных величин. При этом допускаются простые типы, структурные типы и тип массива. К простым относятся встроенные и каталогизированные типы, фиксируемые своими идентификаторами. Структурные типы - это типы обычных записей, ключей виртуальных записей и ключей виртуальных списков. Их указания выглядят как выделенные служебными словами наборы определений списков полей (здесь - рекурсия!). Наконец, диапазоны индексов допускаются лишь числовые и тип элемента массива должен быть простым или структурным типом.

В стандартной нотации определение структуры данных выглядит следующим образом:

ОпрСтруктДанных ::= ОпрГруппыПолей { ОпрГруппыПолей }
ОпрГруппыПолей ::= Имя { "." Имя } ":" УказаниеТипа .

УказаниеТипа ::= ИмяПростогоТипа ";" |
УказаниеСтруктурногоТипа
| УказаниеТипаМассива
| УказаниеТипаПеречисления.

ИмяПростогоТипа ::= ИмяВстроенногоТипа |
ИмяКаталогизированногоТипа.

ИмяВстроенногоТипа ::= BOOLEAN | CHAR | BYTE |
CARDINAL | BITSET | INTEGER | LONGINT | REAL |
INDEFINITE | TEXT | NAME | PICTURE
| BACKGRND.

ИмяКаталогизированногоТипа ::= Имя

УказаниеСтруктурногоТипа ::= ИмяСтруктурногоТипа
ОпрСтруктДанных END ";" .

ИмяСтруктурногоТипа ::= RECORD | LIST OF | BLOCKS OF
| ADDRESS OF.

УказаниеТипаМассива ::= ARRAY Диапазон { "," Диапазон
} OF ТипЭлемента.

Диапазон ::= " [" ЦелоеЧисло ".." ЦелоеЧисло "]" .

ТипЭлемента ::= ИмяПростогоТипа ";" |
УказаниеСтруктурногоТипа.

УказаниеТипаПеречисления ::= =
" (" Имя { "," Имя } ");" .

В этих формулах термин Имя означает любую последовательность символов без пробелов длиной не более 20, не содержащую использованных в контексте разделителей.

Понятие каталогизированного типа и специфичные встроенные типы INDEFINITE, TEXT, NAME, PICTURE, BACKGRND, LIST OF, BLOCKS OF, ADDRESS OF будут разобраны ниже. Прочие встроенные типы определяются следующей таблицей соответствий:

Тип в инструментальной системе	Тип в МОДУЛЕ-2	Тип в Языке C	Длина в байтах
BOOLEAN	BOOLEAN	Char	1
CHAR	CHAR	Char	1
BYTE	BYTE	Char	1
перечисление	перечисление		1
BITSET	BITSET	Unsigned	2
CARDINAL	CARDINAL	Unsigned.	2
INTEGER	INTEGER	int	2
LONGINT	LONGINT	long	4
REAL	LONGREAL	double	8

Чтобы была возможность использовать в разных спецификациях одни и те же структурные типы данных, введен аппарат каталогизированных типов: в библиотеке системы, наряду со спецификациями групп, объектов и приборов, можно заводить в специально выделенной секции спецификации структурных типов данных, тем самым каталогизируя их. Синтаксис спецификации каталогизированного типа таков:

ОписательКаталогТипа. :: = **TYPE** Имя ";"

ОпрСтруктДанных **END** ";"

Смысл употребленных в правой части терминов тот же, что в предыдущей серии определений, причем Имя и есть то

имя, под которым данный тип будет зарегистрирован в библиотеке и под которым его можно упоминать в других спецификациях. Отметим, что приведенное определение рекурсивно, т.е. допускает использование в спецификации каталогизированного типа имен иных типов того же сорта. Циклов в этой рекурсии быть не должно, и система не допустит их, отказываясь компилировать спецификацию, если еще не заведены или не откомпилированы спецификации всех упоминаемых в ней каталогизированных типов.

Примеры спецификаций каталогизированных типов:

```
TYPE VECTOR;
```

```
    X, Y, Z : REAL;
```

```
    END;
```

```
TYPE MATRIX;
```

```
    Rows : ARRAY [1..3] OF VECTOR;
```

```
    END;
```

В номенклатуре встроенных типов есть два графических. Это - PICTURE и BACKGRND. Оба определяют ключи неких картинок, формально являющихся динамическими записями особой структуры. Первый тип относится к маленьким картинкам-пиктограммам. Второй - тип ключа полномерной картинки. Переменные обоих типов предназначаются для графической анимализации моделей. Использование их в программах имитации в качестве фактических параметров соответствующих инструментальных процедур позволяет создавать "мультфильмы" эволюции воспроизводимых процессов. Инициализация этих переменных, включающая и создание ассоциируемых с ними изображений, обычно должна осуществляется через соответствующие графические редакторы, но может выполняться и программно.

Следует упомянуть и три еще не разобранных динамических типа, это типы BLOCKS OF и LIST OF относятся к ключам

блокированных и неблокированных списков из записей одной длины, а тип ADDRESS OF - к ключам одиночных динамических записей. Структура записей всегда фиксируется текстом, следующим за именем типа.

При инициализации все динамические структуры размещаются в копии базы данных, создаваемой системой на время сеанса в виртуальной памяти. По завершении сеанса эту копию можно сохранить, и тогда сохранятся все заведенные данные; тем самым будет обеспечена возможность их использования в последующих сеансах. Пример определения структуры данных:

```
numb : CARDINAL;  
vect : VECTOR;  
cards : LIST OF  
    card : RECORD  
        name : (knave, queen, king);  
        pict : PICTURE;  
END;  
    color : (spades, leaves, diamonds, harts);  
    trump : BOOLEAN;  
END;
```

Операторы коммутации

Согласно принятой концепции часть параметров компонент комплекса может явно моделироваться в других его компонентах, т. е. являться частью фазовых переменных этих компонент. Комплексу также разрешено иметь собственные параметры и фазовые переменные. Для описания возможных связей параметров с фазовыми переменными используются операторы коммутации. Реально стыковка параметров и фазо-

вых переменных, а также проверка возможности такой стыковки осуществляется во время сборки модели.

Общий вид оператора коммутации таков:

ОператорКоммутации ::= {ОпределительДиапазона} АдресПараметра "=" АдресФазовойПеременной ";"

Один оператор определяет один фрагмент связи, причем четко фиксируется направление связи - от контакта, указанного за равенством, к контакту, указанному перед ним. Синтаксис адресов контактов в операторах коммутации определяется формулами вида:

АдресПараметра, ::= {АдресКомпоненты.}ИмяПараметра
{ИндексПараметра}

АдресКомпоненты, ::= ИмяКомпоненты "(" ИндексКомпоненты ")"

АдресФазовойПеременной, ::= {АдресКомпоненты"."}ИмяФазовойПеременной
{ИндексФазовойПеременной}

Если адрес контакта сводится к его индексу, то это значит, что контакт принадлежит внешнему разъему описываемой компоненты модели. ИмяПодгруппы, ИмяОбъекта и ИмяПрибора - суть идентификаторы соответствующих классов компонент. Служебное слово INT помечает внутренние разъемы приборов. Термины ИндексПараметра, ИндексФазовойПеременной формально расшифровываются одинаково:

ИндексПараметра,
ИндексФазовойПеременной ::= "(" ПростоеЦелоеВыражение
)".
ПростоеЦелоеВыражение ::= ЦелоеБезЗнака |

[Целое Знак] "i*" ЦелоеБезЗнака |[Целое Знак]
 ЦелоеБезЗнака "i*" |[Знак] "i*" ЦелоеБезЗнака
 Знак ЦелоеБезЗнака |[Знак] ЦелоеБезЗнака "i*"
 Знак ЦелоеБезЗнака.

Знак ::= "+" | "-".

Во второй формуле все варианты, кроме первого, относятся к случаю, когда записи оператора коммутации предшествует

Определитель диапазона ::= "i=" Целое ".." Целое ":".

Это - конструкция, позволяющая одной записью оператора коммутации определить целую группу фрагментов каналов связи, а каких именно - ясно из контекста. Осталось только подчеркнуть, что указываемые в квадратных скобках номера подгрупп и объектов или приборов являются порядковыми во внутренней индексации той группы или объекта, в чей описатель войдет оператор. Примеры операторов коммутации:

i = 1..10 : ОбъектПервогоТипа (i).МассивПараметров(2*i-1) =

ОбъектВторогоТипа (10-
 i).МассивФазовыхПеременных(2*i);

A1=Спутник.X;

Объект(0).x=x;

Синтаксис описателей комплексов

Двух предыдущих подразделов достаточно, чтобы без подробных дополнительных разъяснений определить принятые правила формирования спецификаций комплексов и ком-

понтент. Формальный синтаксис спецификации комплексов таков:

```
СпецификацияГруппы ::= COMPLEX ИмяКомплекса";"  
    [ПараграфКомпонент]  
    [ПараграфФазовыхПеременных ]  
    [ ПараграфКонстант ]  
    [ПараграфПараметров]  
    [ПараграфКоммутации ].
```

```
ПараграфКомпонент ::= COMPONENTS  
    ПереченьСоставляющих; { ПереченьСоставляющих }  
    {END";"}
```

```
ПараграфФазовыхПеременных ::= PHASE  
    ОпрСтруктДанных  
    {END ";"}
```

```
ПараграфПараметров ::= PARAMETERS  
    ОпрСтруктДанных  
    {END ";"}
```

```
ПараграфКонстант ::= CONST  
    ОпрСтруктДанных  
    {END ";".}
```

```
ПараграфКоммутации ::= COMMUTATION  
    ГруппаКоммутации { ГруппаКоммутации. } {END ";".}
```

Во всех параграфах, кроме последнего в комплексе, **END'ы** необязательны. Заголовок следующего параграфа автоматически означает конец предыдущего.

Смысл использованного здесь термина ОпрСтруктДанных разобран в предыдущем разделе, а несложные формулы

определений других новых терминов выглядят следующим образом:

ПереченьСоставляющих ::= ИмяКомпоненты " (" ЧислоЭкземпляров ")"

{",", " ИмяКомпоненты "(" ЧислоЭкземпляров ")" }";".

ГруппаКоммутации ::= [ОпределительДиапазона]

ОператорКоммутации.

В этих формулах ИмяКомплекса и ИмяКомпоненты суть имена классов компонент соответствующего типа, а формально – идентификаторы длиной не более 20 символов; ЧислоЭкземпляров – положительное целое, равное числу составляющих указанного перед скобкой класса; ОпределительДиапазона, ОператорыКоммутации - термины из предыдущего раздела.

Синтаксис описателей методов

Методы – это или элементы процессов, или же методы вычисляющие наступление событий. Это те части модели, которые могут вызываться как локально, так и распределенно. Считается, что в качестве параметров методу передается некая запись, тип которой описан, и она же возвращается методом. Так как тип этой записи, в особенности для удаленных методов, определяется разработчиком метода, а не разработчиком модели, возникает вопрос о коммутации полей параметров метода с полями фазовых переменных и/или параметров компоненты. Кроме того, методы реализующие элементы процессов различаются по отношению с модельному времени на

- сосредоточенные – происходящие мгновенно,

- распределенные – занимающие не менее одного модельного такта и дающие определенный результат своего выполнения в виде изменений внутренних переменных модели в конце каждого такта,
- условно-распределенные – занимающие не менее одного модельного такта, но дающие результат своего выполнения в виде изменений внутренних переменных модели лишь при полном завершении выполнения.

ОписательМетода ::= METHOD ИмяМетода ":" ТипМетода
";"

[ПараграфПараметровМетода]

ТипМетода ::= **FAST** | **CONV** | **SLOW** | **EVENT**.

По умолчанию тип метода, реализующего элемент процесса, считается **SLOW**, и в этом случае явное его указание разрешается опустить.

Синтаксис параграфа параметров метода такой же, как у параграфа фазовых переменных описателя комплекса. Поскольку это единственный параграф описателя, он обязан закончиться ключевым словом **END**.

Синтаксис описателей компонент

Формула спецификации компоненты такова:

ОписательКомпоненты ::= **COMPONENT** ИмяКомпоненты ";

[ПараграфФазовыхПеременных]

[ПараграфПараметров]

[ПараграфКонстант]

[ПараграфМетодов]

[ПараграфКоммутации]

[ПарграфСобытий]

[ПарграфПереключателей].

Как и в случае комплексов, **END** обязателен лишь в самом последнем параграфе. В параграфе методов спецификации компоненты одним или несколькими процессами перечисляются все выполняемые в данном процессе методы, начиная с корневого, с которого по умолчанию будет начинаться модельная жизнь каждого процесса. Имена методов должны быть уникальными в пределах спецификации: нельзя одинаково назвать два разнотипных метода одной компоненты, но использование одного имени в спецификациях разных компонент не возбраняется. Формально синтаксис параграфа методов определяется так:

```
ПараграфМетодов ::= ПотокМетодов |  
    | ПараграфМетодов ПотокМетодов {END  
    "};
```

```
ПотокМетодов ::= ИмяМетода { ", " ИмяМетода  
    } "};
```

Здесь ИмяМетода - идентификатор длиной не более 20 символов.

Параграф коммутации синтаксически не отличается от параграфа коммутации компонент комплекса. Содержательный же его смысл в том, что параметры методов достаются разработчику модели от разработчиков методов такими, какими они были удобны последним, в частности множество параметров метода может быть существенно уже множества фазовых переменных, параметров и констант компоненты, а может и совпадать или даже оказаться шире последнего. В лю-

бом случае параграф призван указать, каким параметрам метода соответствуют те или иные фазовые переменные, параметры и константы компоненты.

ПараграфКоммутации ::= **COMMUTATION**
ГруппаКоммутации { ГруппаКоммутации. }
{END ";"}.
ГруппаКоммутации ::= [ОпределительДиапазона]
ОператорКоммутации.

Параграф событий определяет набор событий, связанных с данной компонентой. Определяется идентификатор события и условие его наступления.

ПараграфСобытий ::= ОписаниеСобытия { ";"
ОписаниеСобытия } ";" **{END ";"}**.

ОписаниеСобытия ::= ИмяСобытия ":" МетодВычисляющийСобытие ";".

Параграф переключателей определяет автоматную функцию процесса. В нем для каждого элемента каждого процесса должны быть перечислены все разрешенные переходы, т.е. названы все элементы, на которые прибор может переключаться по завершении данного, и указаны события, в зависимости от которых реализуется то или иное переключение. Формат параграфа таков:

ПарграфПереключателей ::= **SWITCHES** АвтоматнаяФункция
{END ";"}
АвтоматнаяФункция ::= ПереключенияЭлемента
{ ПереключенияЭлемента }.
ПереключенияЭлемента ::= ИмяЭлемента ":" СписокПереходов.

СписокПереходов ::= { ИмяЭлемента "," ИмяСобытия
"."}ИмяЭлемента ";"

Понятно, что перед двоеточием во второй формуле стоит имя того элемента, переходы с которого определяет следующая за двоеточием конструкция. В последней перечисляются возможные переключения. Они упорядочиваются по важности событий, причем завершается перечень именем элемента, при котором не стоит никакого события. Это значит, что если не реализуется ни одно из вошедших в СписокПереходов событий, то произойдет переключение на элемент с указанным именем.

Компиляция описателей компонент модели

После того как имеется описание модели и всех ее компонент, можно перейти к компоновке модели. Первый этап компоновки модели – компиляция описателей. Компиляция описателя – это получение из него одной или нескольких незаполненных таблиц для будущей базы данных модели. Например, в результате компиляции описания типа

```
TYPE VECTOR;  
    X, Y, Z : REAL;  
END;
```

в базе данных модели в таблице «Типы_модели» появится еще одна строка вида:

Типы_модели

Номер_типа	Имя_типа	Размер_типа
...
n	VECTOR	24

кроме того, в таблице «Поля_типов» добавятся три новых строки:

Поля_типов

Ключ поля типа	Нмер типа	Номер поля	Имя поля	Тип поля	К-во экземпляров
	
k	N	1	X	REAL	1
k+1	N	2	Y	REAL	1
k+2	N	3	Z	REAL	1

Таблицы «Типы_модели» и «Поля_типов» связаны отношением «один ко многим». Заметим, что поле «Размер_типа» в таблице «Типы_модели» может остаться незаполненным, если типы полей компилируемой записи сложнее, чем встроенные. В последнем случае это поле либо будет заполнено в процессе компоновки компоненты, содержащей этот тип, если удастся пройти до конца (до встроенных типов, являющихся терминальными элементами) всю цепочку взаимных ссылок типов, либо компоновка закончится аварийно, например, если какой-то упомянутый в описании поля тип не будет найден в таблице «Типы_модели», либо если цепочка взаимных ссылок типов зациклится. Чтобы не придумывать для встроенных типов каких-то специальных механизмов узнавания, а пользоваться для всех типов одним и тем же алгоритмом, можно считать, что все встроенные типы изначально занесены в таблицу «Типы_модели» и отсутствуют в таблице «Поля_типов».

В результате компиляции описателя компоненты добавляется очередная строка в таблицу «Компоненты_модели»:

Компоненты_модели

Но-мер	Имя ком-	Фаза	Пара-метры	Кон-станты	Про-цессы	Число про-

ком- по- нен- ты	по- нен- ты				начало	цессов
...	
n	Имя	n_Ph	N_Par	n_Con	n_Pr_b	n_Pr

Здесь n – номер компоненты – ключевое поле в таблице компонент модели, n_Ph, n_Par, n_Con – номера типов фазовых переменных, параметров и констант модели в таблице «Типы_модели», n_Pr_b – номер первого процесса компоненты, а n_Pr – количество процессов компоненты в таблице «Процессы_модели». При этом предполагается, что все записи процессов компоненты идут подряд в таблице «Процессы_модели». Соответственно в таблице «Процессы_модели» появляется n_Pr новых записей, соответствующих процессам компилируемой компоненты:

Процессы_модели

Номер процесса	Корневой элемент

Также будут добавлены новые строки в таблицы «Методы_модели»,

Номер метода	Имя метода	Номер процесса	Парамет- ры метода	Тип мето- да	Адрес метода

Чтобы получить, например, список имен и адресов методов m-го процесса, необходимо выполнить SQL-запрос:

```
SELECT [Имя элемента], [Адрес элемента] FROM
[Методы_модели] WHERE [Номер процесса]=m.
```

Коммутация методов должна быть отражена в следующей таблице:

Коммутация_методов

Ключ коммутатора методов	Номер метода	Поле параметра метода	ФПК (фаза-парам.-конст.)	Поле компоненты

«Переключатели_модели»

Ключ переключателя	Важность переключателя	Номер процесса	Номер элемента	Номер события	Элемент перехода

Поле «Важность переключателя», вообще говоря, избыточно. Можно было бы договориться, что переключатели добавляются в таблицу «Переключатели_модели» в порядке важности, так как они фигурируют в параграфе «переключатели» описателя компоненты. Однако во избежание путаницы в этом важном вопросе, представляется целесообразным добавить такое поле. Чтобы получить, например, сортированный в порядке важности список событий перехода и следующих элементов для элемента n, m-го процесса, необходимо выполнить SQL-запрос:

```
SELECT [Номер события], [Элемент перехода]
FROM [Переключатели_модели] WHERE
[Номер процесса]=m AND [Номер элемента]=n
ORDER BY [Важность переключателя].
```

Таким образом, компиляция описателя компоненты может вызвать пополнение следующих таблиц базы данных:

- Типы_модели,

- Поля_типов,
- Компоненты_модели,
- Процессы_модели,
- Методы_модели,
- Коммутация_методов,
- Переключатели_модели.

Теперь остановимся на том, как компилируется описатель комплекса. В рамках предлагаемой концепции построения инструментальной системы распределенного моделирования, понятие комплекса является вспомогательным, удобным для построения сложных иерархических моделей. При компиляции комплекс преобразуется в «комплекс как компоненту» путем объединения констант, фазовых переменных, процессов, элементов, событий и наконец, параметров. При этом из объединенных параметров исключаются те, что участвуют в коммутации компонент [1]. При исключении параметра, нужно проследить с какими методами он коммутировал, и заменить в строках таблицы коммутации методов комплекса как компоненты поля связанные с этим параметром на поля, соответствующей фазовой переменной.

На первом этапе компиляции описателя комплекса, добавится строчка в таблицу «Комплексы_модели»

Комплексы_модели

Ключ комплекса	Имя комплекса	Фазовые переменные	Параметры	Константы

Перечень компонент компилируется в несколько строк таблицы «Компоненты_комплексов».

Компоненты_комплексов

Ключ компонент комплексов	Номер комплекса	Номер компоненты	Количество экземпляров

Параграф коммутации компилируется в строки таблицы коммутации компонент:

Коммутация_компонент

Ключ коммутатора	Номер комплекса	Номер вх. компонента	Экземпляр компонента	Поле параметра	Номер исходной компоненты	Экземпляр компоненты	Поле фазовой переменной

Далее на основе этих таблиц можно строить «комплекс как компоненту», объединяя с учетом количества экземпляров фазовые переменные, константы, параметры, процессы, элементы, события.

Выбрать параметры, подлежащие исключению можно сделав запрос

```
SELECT * FROM [Коммутация_компонент] WHERE [Номер комплекса] = n
```

Выбрать методы, подлежащие перекоммутации, можно по запросу:

```
SELECT [Номер метода] FROM [Коммутация_методов] WHERE  
[ФПК] = "параметры" AND [Поле компоненты] = ИмяПоля.
```

Наконец, поля фазовых переменных, на которые нужно заменить поля параметров в таблице «Коммутация_методов», также находятся в таблице первого из запросов.

Компоновка модели

На стадии компиляции описателей модели формируется, как это было показано выше, база данных, описывающая состав компонент модели и связи между ними. Задача этапа компоновки – проверить полноту и непротиворечивость системы этих описаний и сгенерировать на их основе рабочую базу данных для последующего проведения имитационных экспериментов.

Исходным материалом для компоновки модели является результат компиляции компоненты или «комплекса как компоненты», а именно таблицы

1. Типы_модели,
2. Поля_типов,
3. Компоненты_модели,
4. Процессы_модели,
5. Методы_модели,
6. Коммутация_методов,
7. Переключатели_модели.

На основе первой и четвертой из них строятся две новые таблицы: по имеющимся описаниям типов переменных выделяется реальное пространство или для всего имитационного эксперимента, или же для нулевого момента времени, с последующим выделением пространства для очередного шага имитации. Заполнение базы данных начальными условиями

осуществляется вручную или же с помощью специальной программы, в данной работе мы не останавливаемся на технике заполнения базы данных модели начальными значениями, хотя этот процесс и может оказаться нетривиальным. Также таблица «Процессы модели» разворачивается в модельном времени, и на каждом шаге моделирования в ней будет указываться текущий элемент процесса.

Работа модели во время эксперимента

Предположим, что мы находимся в начале такта имитации. Задан некий стандартный шаг времени Δt .

- Просматривая таблицу процессов, выполняем текущие сосредоточенные элементы, передавая им параметры в соответствии с таблицей коммутации методов. Возвращенные значения параметров присваиваем соответствующим фазовым переменным. В соответствии с таблицей переключателей, вычисляем следующие элементы.
- Повторяем предыдущий шаг, пока имеются сосредоточенные элементы.
- Выполняем со стандартным шагом Δt распределенные и условно-распределенные элементы. Вычисляем наступление событий, связанных с очередным элементом. В частности, такими событиями могут быть и заранее запланированные с помощью специальных системных средств, как в [2], моменты окончаний элементов. Как указывалось в [1], наступление события есть обращение в ноль соответствующей функции. Методы событий реализуют именно такие функции. Смена функцией события знака на концах отрезка Δt говорит о наступлении события внутри этого отрезка. Далее с помощью того или иного метода аппроксимации с приемлемой

точностью находится момент наступления события. Минимум из этих времен по всем элементам даст продолжительность следующего шага имитации, после которого для закончившихся элементов в соответствии с таблицей переключателей и наступившими событиями вычисляются последующие.

- Модельное время переводится на вычисленную продолжительность прошедшего шага имитации.

Мы оказываемся в начале нового такта имитации, который снова пытаемся пройти, исходя из стандартного шага Δt .

Работа с моделью во время имитационного эксперимента

Предполагается возможность в любой момент приостановить имитационный эксперимент, после чего исследователь может работать с характеристиками модели средствами применяемой СУБД. После работы с базой данных возможно возобновление приостановленного имитационного эксперимента.

4. Компьютерная реализация макета распределенной эколого-демографо-экономической модели в режиме распределенных вычислений

Макет распределенной эколого-социально-экономической модели, описанной в разд. 1 и 2 реализован на нескольких (в настоящее время до семи) компьютерах локальной сети или глобальной сети Интернет (в последнем случае, эти компьютеры должны иметь маршрутизируемые в Интернет IP-адреса).

Технической основой взаимодействия распределенных компонент модели является разработанная в ИСА РАН архитектура IARnet, описанная в [18]. Реализация, о которой здесь идет речь, выполнена пока без полного использования описанной в разд. 3 технологии. В частности, отсутствует описание моделируемого комплекса на специализированном непроцедурном языке, позволяющем автоматически сгенерировать его базу данных.

Однако как следует из приводимого ниже описания распределенной реализации, основная идея, заложенная в разбиение модели на распределенные компоненты, полностью соответствует представленной в разд. 3 технологии. Она основана на объектно-событийном представлении, позволяющим выделить ряд классов объектов (компонент в терминах технологии, изложенной в разд. 3), которые достаточно независимы друг от друга, т. е. взаимодействуют друг с другом лишь в определенные моменты жизни модели. Экземпляры этих классов (компонент) могут быть реализованы на отдельных компьютерах. Ниже кратко описываются эти классы:

- Класс "информационный сервер" присутствует в единственном экземпляре. Этот объект собственно, является не частью модели, а надстройкой над ней, осуществляющей взаимодействие пользователя с моделью, начиная от создания реализации модели для конкретного имитационного эксперимента и кончая заданием управлений функционированием стран и отображением результатов имитации. Данный объект реализован как HTTP-сервер, основу которого составляют странички, написанные на языке HTML, усиленные клиентскими скриптами, написанными на языке JavaScript, и серверными скриптами, написанными на языке Perl. Стран-

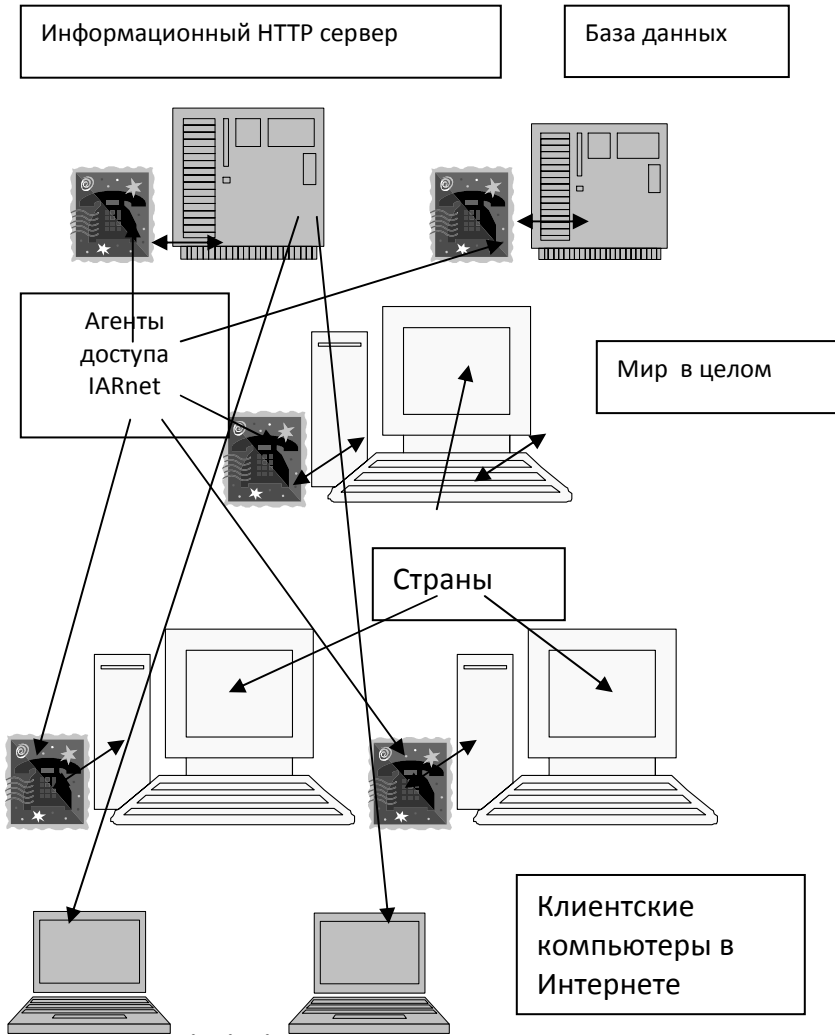
нички информационного сервера отражают состояния демографии, экономики и экологии участвующих в имитационном эксперименте стран. Для связи с остальными распределенными компонентами на информационном сервере, в соответствии с архитектурой IARnet постоянно работает **агент доступа**, обеспечивающий такую связь.

- Класс "страна", который в макете представлен несколькими экземплярами (от одного до четырех). У объектов этого типа два основных метода – это, во-первых, метод-конструктор, позволяющий создать новую страну, т. е. создать и заполнить начальными данными соответствующие информационные структуры. Данный метод поддержан на информационном сервере соответствующим серверным скриптом, с помощью которого пользователь в начале имитационного эксперимента может вызвать метод-конструктор и создать страну – участницу эксперимента. Второй метод – это шаг моделирования, т. е. метод, позволяющий модельной "стране" прожить очередную единицу модельного времени. Этот метод на информационном сервере также поддержан серверным скриптом, позволяющим лицу, принимающему решения за данную страну, задать управления на очередной модельный год. Считается, что могут быть реализованы различные вычислительные алгоритмы, связанные с методами стран. В данном макете реализованы три таких алгоритма, отражающие тот факт, что, создавая в начале эксперимента очередную страну-участницу, можно выбирать один из трех возможных типов ее экономики. Экземпляры объекта "страна" могут быть размещены на отдельных компью-

терах сети. Они также должны быть снабжены агентами доступа.

- Класс "хранилища демографической информации" представлен в макете одним экземпляром, в котором может храниться демографическая информация трех различных типов, т. е. создавая страну, ее начальную демографическую ситуацию, можно выбирать из трех качественно различных вариантов. В этом хранилище содержится демографическая информация, относящаяся к соответствующей стране, и оно также оснащено агентом доступа к этой информации в соответствии с архитектурой IARnet. Хранилище демографической информации также может располагаться на отдельном компьютере.
- Класс "мир" представлен в макете единственным экземпляром. Метод этого класса моделирует взаимодействия стран такие как, например, миграция населения, импорт/экспорт товаров. Объект "мир" также должен иметь агент доступа, и может располагаться на отдельном компьютере.

Графически архитектура макета распределенной эколого-социально-экономической модели представлена на рис.1.



..
 Рис. 1. Структура макета эколого – демографо -экономической распределенной модели

На рис. показаны только компоненты самого макета. Не показано взаимодействие лиц, принимающих решение за моделируемые страны, с информационным сервером, осуществляющим пользовательский интерфейс с макетом. Такое взаимодействие может быть осуществлено с любой рабочей станции, имеющей выход в Интернет и браузер, графически схема этого взаимодействия показана на рис. 2.

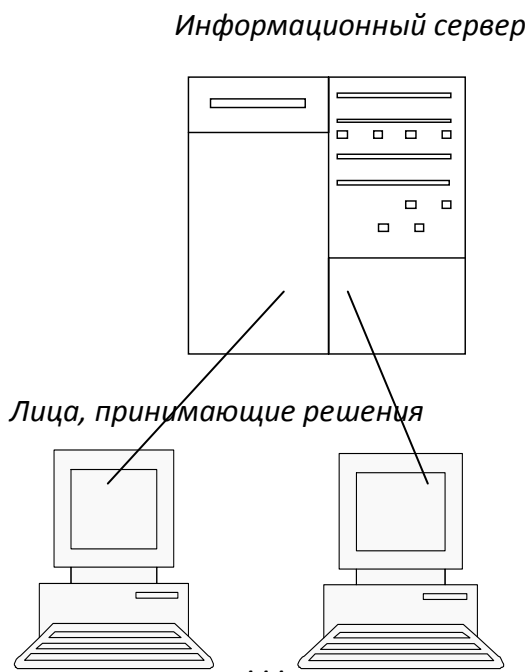


Рис. 2. Схема взаимодействия игроков с моделью.

Макет распределенной эколого-социально-экономической модели может быть реализован на семи различных компьютерах. Каждый информационно-алгоритмический ресурс, расположенный на отдельном компьютере, снабжен соответствующим агентом доступа.

Описание агентов доступа

Общая структура распределенной имитационной модели включает четыре типа агентов доступа: агент доступа к демографической базе данных, агент доступа к модели страны (их может быть до четырех экземпляров), агент доступа к модели мира и агент доступа к информационному серверу. Каждый тип агента имеет уникальный интерфейс, который реализует удаленные методы и, тем самым, обеспечивает возможность взаимодействия агентов между собой.

Агент доступа к демографической базе данных обеспечивает передачу данных требуемого типа из базы данных и в базу данных.

Агент доступа к модели страны (их может быть от одного до четырех) реализует следующие функции: инициализацию (создание) экземпляра страны в соответствии с выбранным типом, передачу заданного пользователем управления; передачу текущего года моделирования, выполнение очередного шага моделирования для страны, генерацию и передачу графической информации о состоянии страны.

Агент доступа к модели мира (взаимодействия между странами) реализует следующие функции: осуществляет передачу параметров от стран до их взаимодействия и странам после него, осуществляет очередной шаг для взаимодействия между странами.

Агент доступа к информационному серверу в Интернете является центральным связующим звеном модели. Он реализует три основные функции: глобальная инициализация модели; шаг для модели в целом; освобождение агентов задействованных в модели, для их повторного использования.

Схема взаимодействия агентов доступа и информационно-алгоритмических ресурсов выглядит, как показано на рис. 3.

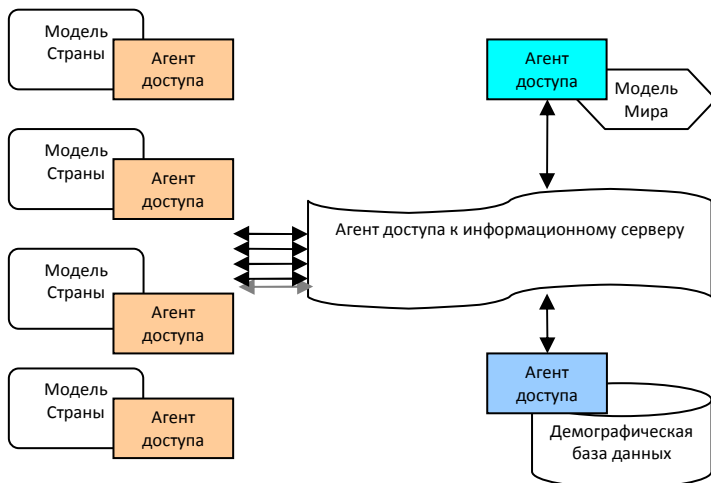


Рис. 3. Схема взаимодействия агентов доступа и информационно-алгоритмических ресурсов

1. Инициализация.

При инициализации выполняется следующая последовательность действий:

- Вызов метода АДкДБД
`void getData(in Identifier m_T,out Parameters m_param)`
- Вызов метода АДкС
`void init(in Identifier m_TYPE,in Parameters m_param)`

2. Шаг модели.

Шаг модели включает следующую последовательность действий:

- Вызов метода АДКИС
void getCtrParam(in Identifier m_C,out Parameters m_param)
- Вызов метода АДКС
void setCtrParam(in Parameters m_param)
- Вызов метода АДКС
void step()
- Вызов метода АДКС
void getParam(out Parameters m_param)
- Вызов метода АДКМ
void setParam(in Identifier m_C,in Parameters m_param)
- Вызов метода АДКМ
void step()
- Вызов метода АДКМ
void getParam(in Identifier m_C,out Parameters m_param)
- Вызов метода АДКС
void setParam(in Parameters m_param)
- Вызов метода АДКС
void getPicture(out Parameters m_param)
- Вызов метода АДКИС
void setPicture(in Identifier m_C,in Parameters m_param)

Дальше цикл повторяется.

Список сокращений:

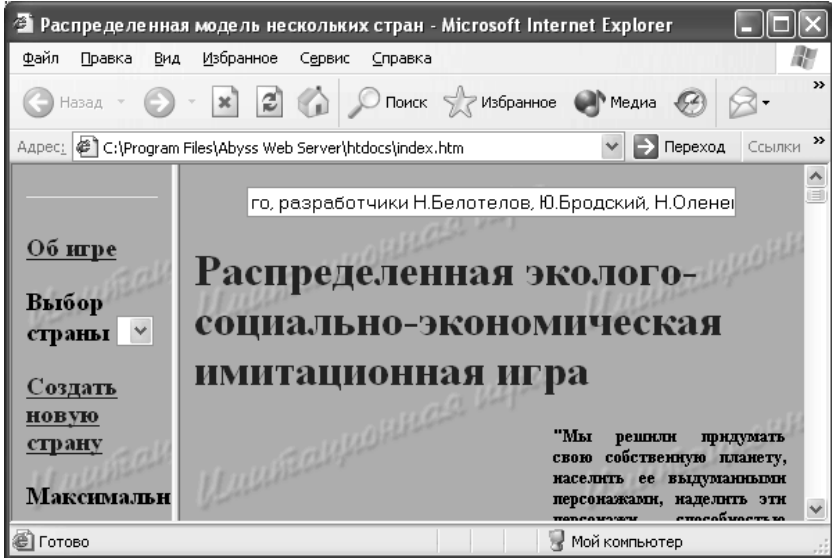
1. АД – агент доступа.

2. ИС – информационный сервер.
3. С – страна.
4. ДБД – демографическая база данных.
5. М – мир.

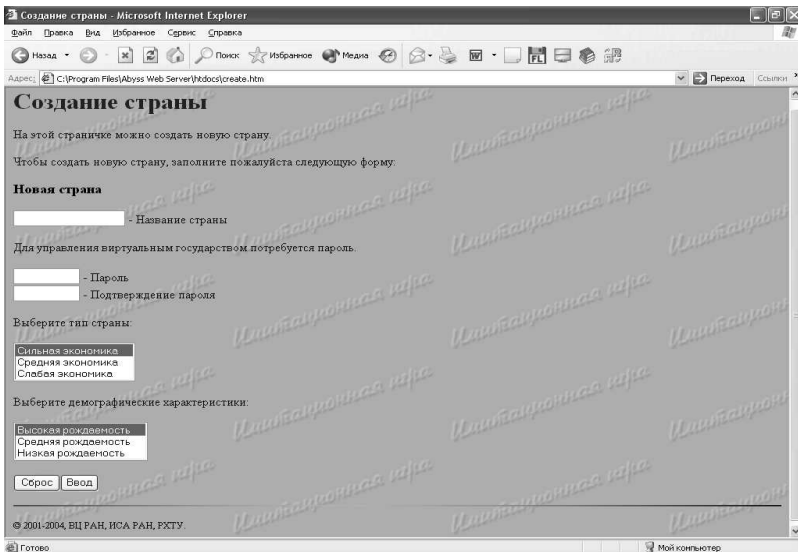
Интерфейс модели

Основой пользовательского интерфейса модели является сайт в Интернете, посвященный данной модели и реализованный на информационном сервере. В настоящее время адрес этого сайта – <http://simul.ccas.ru/Distr>. На этом сайте можно ознакомиться с проектом создания распределенной имитационной модели эколого-социально-экономических процессов, с подробным описанием всех компонент модели, правилами и инструкциями по работе с ней.

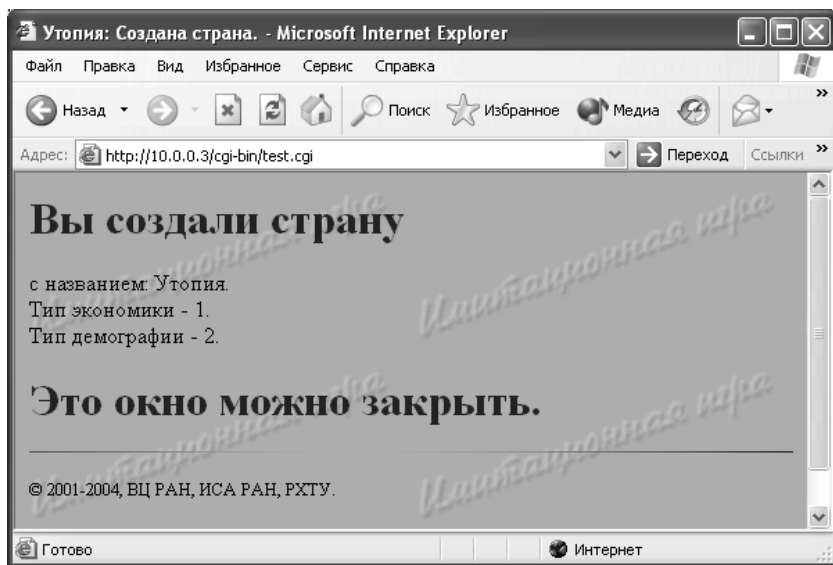
В начальный момент в модели стран еще нет, их предстоит создать. Страничка макета в Интернете имеет вид, показанный на рисунке ниже. Из всех возможностей управления активна лишь возможность создать новую страну.



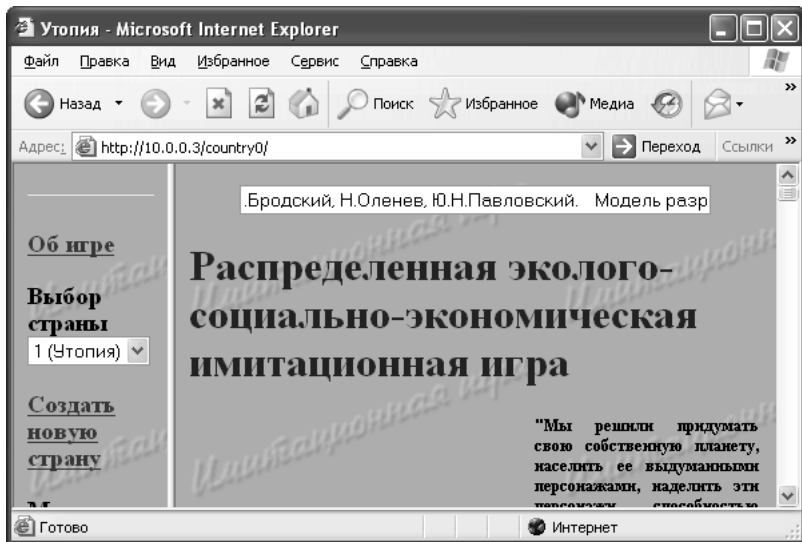
Если выбрать эту возможность, откроется окно создания страны:



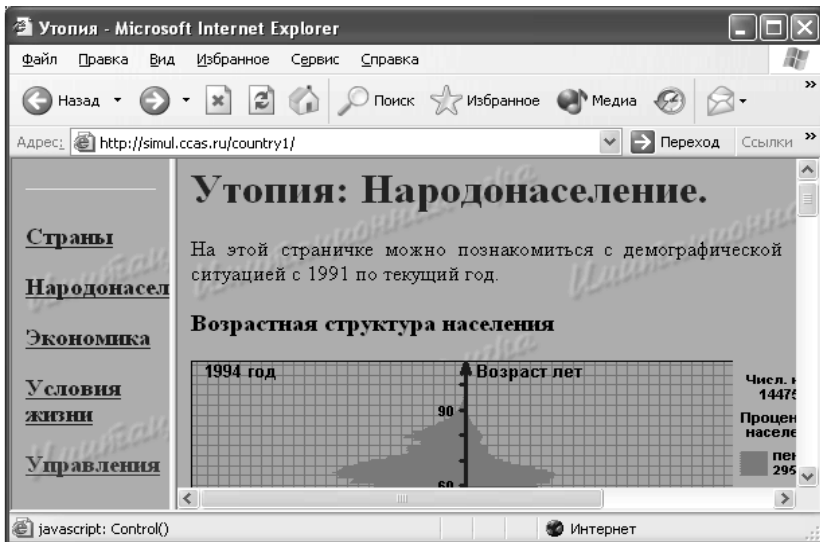
В окне создания страны нужно задать имя страны (под этим именем она будет далее известна на протяжении этого имитационного эксперимента), затем нужно задать пароль и подтвердить его во избежание случайной ошибки. По этому паролю в дальнейшем ходе имитационного эксперимента, можно будет управлять данной страной, т. е. для успешного задания управлений за эту страну необходимо указать пароль, заданный во время создания страны. Далее нужно задать один из трех возможных типов экономики и один из трех возможных типов начальной демографической ситуации. После ввода перечисленных выше сведений появится окно подтверждения создания страны:



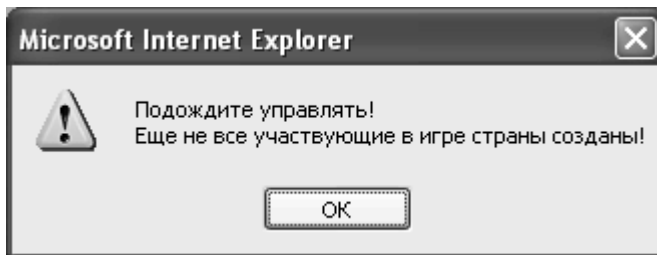
В главном окне макета также появится название созданной страны:



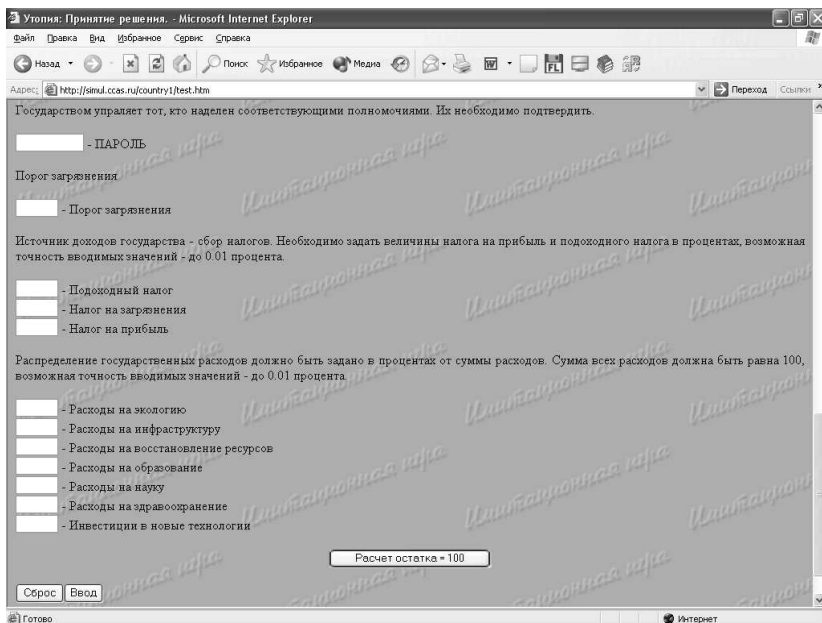
Начать имитационный эксперимент можно только после того, как созданы все четыре участвующие в нем страны. Для этого нужно выбрать нужную страну в меню "выбор страны", затем в меню страны выбрать пункт "управления".



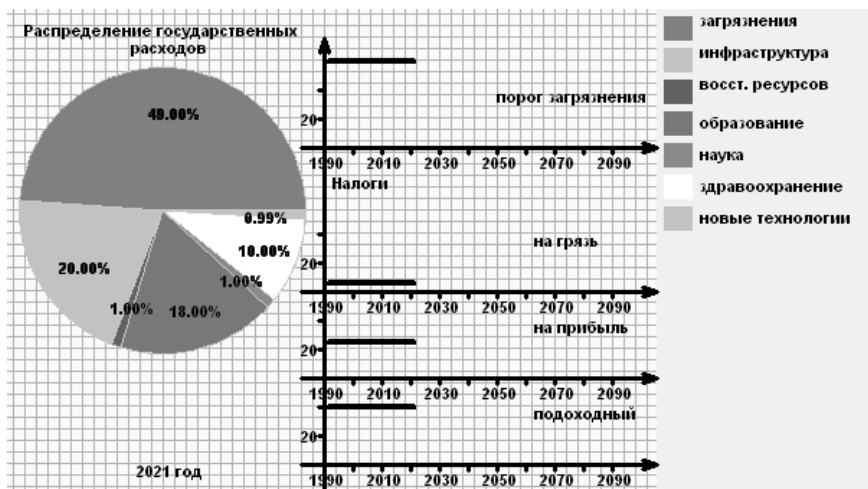
Если попытаться выбрать пункт "управления" какой-либо страной до того, как созданы все четыре страны, появится следующее окно аварийной диагностики:



Если же все страны созданы, появится консоль управления страной:



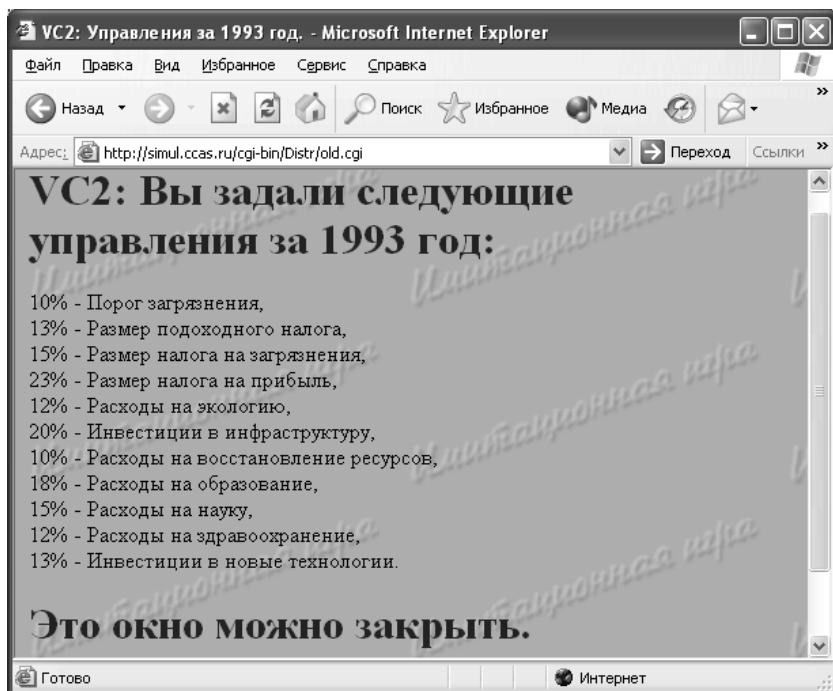
На этой консоли можно познакомиться с ранее задававшимися управлениями (историей управлений, она показана на следующем рисунке)



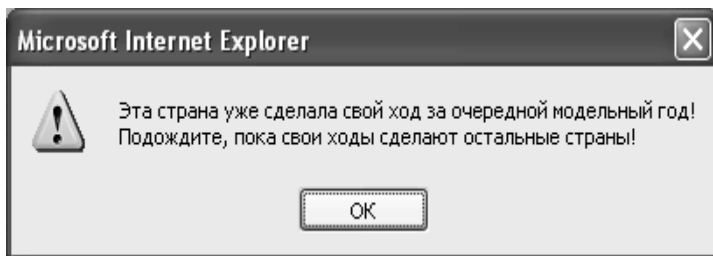
и задать новые управления. Для этого нужно, во-первых, знание пароля, который был задан, когда страна создавалась. Далее задается порог загрязнения, т. е. государство задает границу между экологически чистым и грязным продуктом (что считается чистым, что грязным). Далее государство задает источник своих доходов (величины налогов – подоходного, на загрязнения и на прибыль). Далее государство распределяет свои доходы на семь статей расходов:

1. Экология.
2. Развитие производства.
3. Восстановление ресурсов.
4. Образование.
5. Наука.
6. Здравоохранение.
7. Новые технологии.

Расходы задаются в процентах от бюджета, для удобства расчетов имеется калькулятор остатка. После задания управлений появляется окно подтверждения:

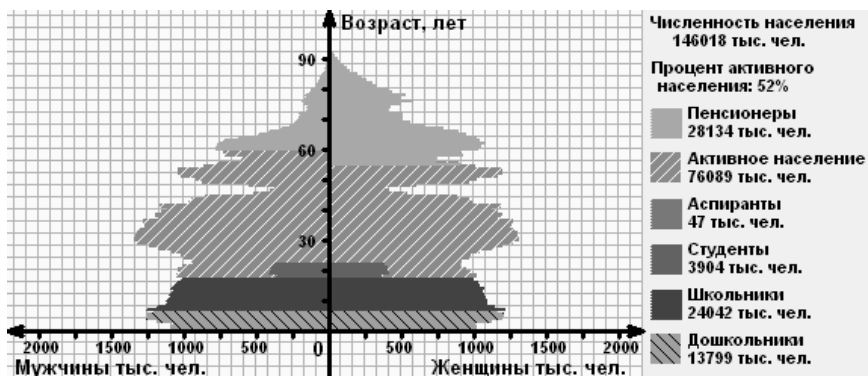


После того как управления заданы за все участвующие в эксперименте страны, проходит очередной модельный год, вызываются соответствующие методы, на информационном сервере появляются соответствующие картинки, снова появляется возможность задавать управления. Если попытаться снова войти в консоль управления, до того как зададут управления остальные участники, появится следующая аварийная диагностика:

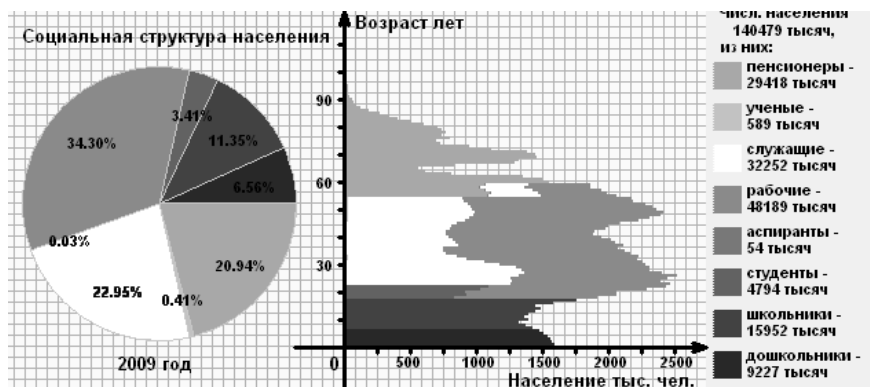


Во время имитационного эксперимента можно следить за демографической, экономической и экологической ситуацией любой из стран на соответствующих страницах информационного сервера. На демографических страничках в виде диаграмм и графиков отображены возрастная и социальная структуры населения, также динамика народонаселения по группам: все население, активное население, пенсионеры, безработные, - и уровень безработицы по социальным группам: рабочие, служащие, научные работники.

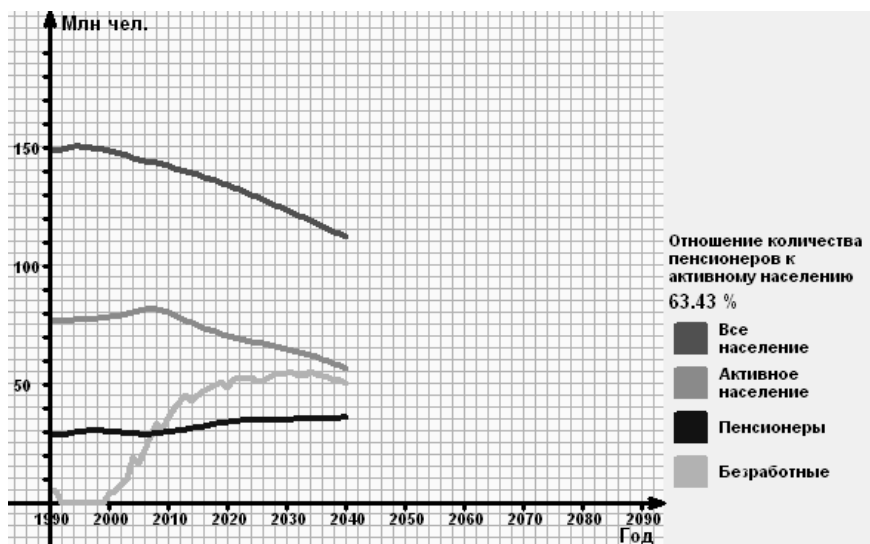
Возрастная структура населения



Социальная структура населения



Динамика народонаселения

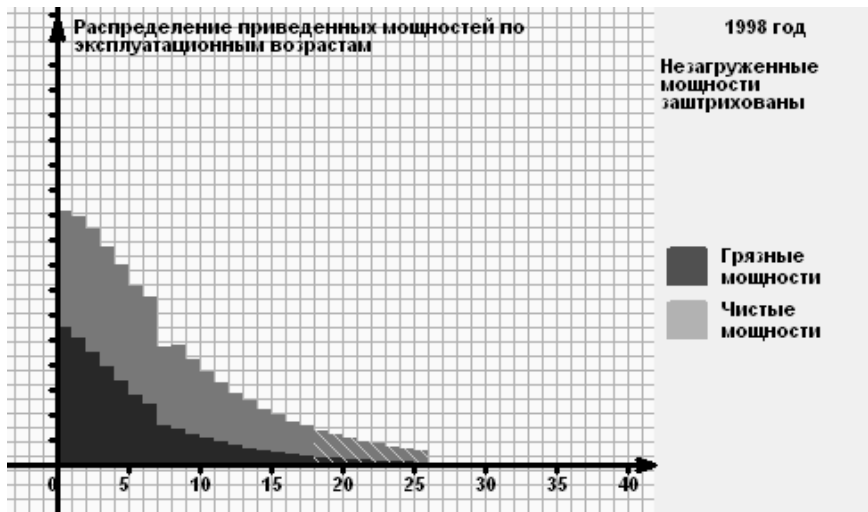


Безработица по социальным группам

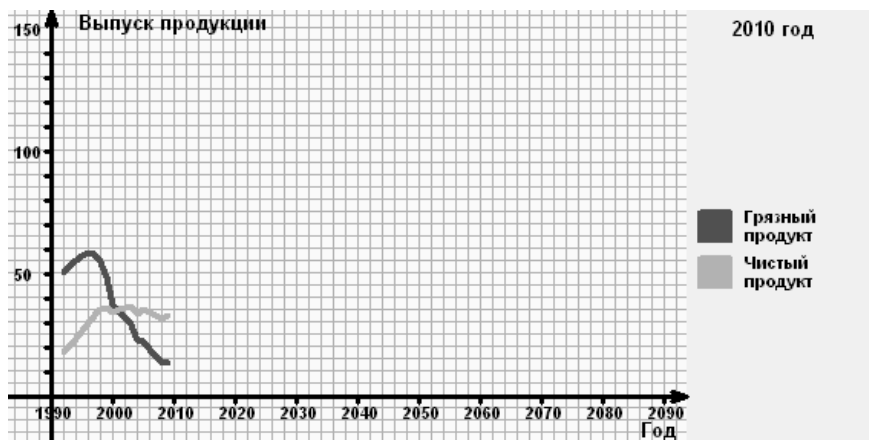


На экономических страничках отображаются возрастная структура производственных мощностей, выпуски продукции и цены на продукцию.

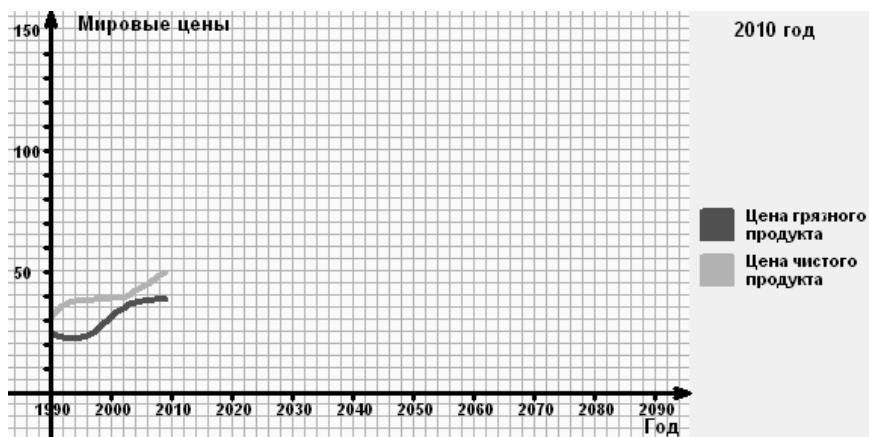
Распределение мощностей по эксплуатационным возрастам



Выпуск продукции

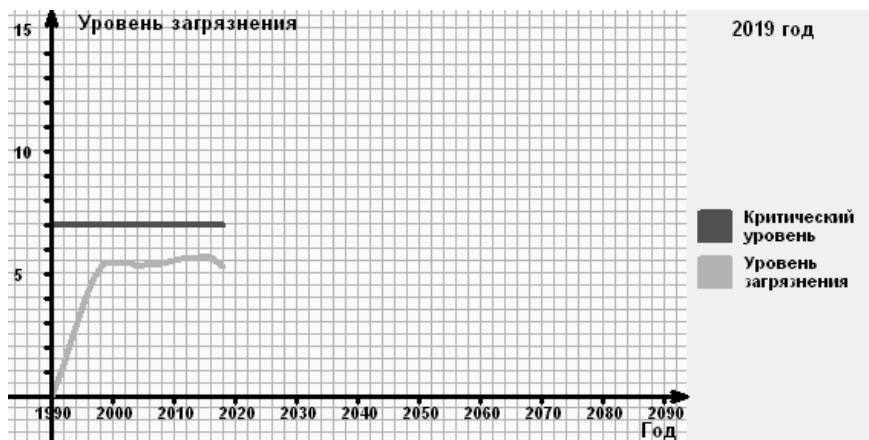


Цены на продукцию



На страничках, посвященных условиям жизни в странах, отражаются экологические условия и уровень потребления населения.

Уровень загрязнения окружающей среды



Потребление на душу населения



Организация имитационного эксперимента

В начальный момент исходные значения характеристик стран задаются участниками имитационного эксперимента и методом-конструктором класса "страна". Метод объекта "мир" устанавливает начальное модельное время и вызывает программу визуализации, которая переводит наблюдаемую часть фазовых переменных игры в вид, принятый для отображения хода игры на Интернетовском сайте (диаграммы, графики, таблицы, числа, картинки...), и засылает этот материал на сайт игры, - начинается первый такт имитации. В дальнейшем участники эксперимента могут сделать свой ход, посетить сайт игры. Метод объекта "мир" отслеживает ходы игроков.

Следующий такт игры (перевод часов модельного времени) наступает, когда все участники эксперимента сделали свой ход (либо он может быть привязан к физическому времени и наступать раз в его единицу: час, сутки, неделю... . В этом случае, если кто-то из участников эксперимента не сделал своего хода, метод объекта "мир" определенным образом назначает ему управление, например прошлое, или усреднение прошлых управлений, либо назначает некоторое управление, специально заданное для таких случаев).

При наступлении очередного такта эксперимента метод объекта "мир" снова ожидает, пока методы объектов-стран пересчитают фазовые переменные в соответствии с заданными оператором управлениями, затем переводит модельные часы и вызывает программу визуализации, - наступает следующий такт жизни модели.

Подготовка распределенного имитационного эксперимента

Информационный сервер модели

В настоящее время для отображения страниц информационного сервера модели в Интернете используется свободно распространяемый WWW-сервер Abyss Web Server X1 версии 1.2.1.0 фирмы Aprelium Technologies.

Для успешного функционирования информационного сервера необходима определенная конфигурация переменных и каталогов, находящихся в папке **htdocs** – корневом каталоге WWW-сервера. В этой папке должна находиться переменная **server.js**, в которой записан IP или DNS адрес информационного сервера. Синтаксис записи - синтаксис Java Script:

```
var server="193.233.208.15"; или var server="simul.ccas.ru";;
```

Все остальные переменные и каталоги, относящиеся к распределенной модели, находятся в каталоге **Distr** корневого каталога **htdocs**. (Адрес макета в Интернете-<http://simul.ccas.ru/Distr>).

Каталог Distr

В этом каталоге расположены десять подкаталогов с именами **country0**,...,**country9**, в которых находится информация, относящаяся к странам, участвующим в эксперименте.

Далее в каталоге *Distr* находятся агент доступа *AA_Main.exe* к информационно-алгоритмическому ресурсу «информационный сервер модели» и командный файл *Main.bat* для активизации этого агента доступа, а также файл привязки к службе имен CORBA – *ns.ior*.

Для правильной работы распределенной модели необходимо, чтобы этот агент доступа был активизирован вручную (запущен командный файл *Main.bat*), до начала работы с моделью.

Далее в каталоге *Distr* находятся переменные *countr.js* и *MaxCountr.js* в которых хранится текущее число стран и максимальное число стран, участвующих в имитационном эксперименте. Синтаксис записи - синтаксис Java Script:

***var countr=4;* и соответственно *var MaxCountr=4;* Значения этих переменных могут отличаться друг от друга (при этом *countr < MaxCountr*) только на этапе создания стран, участвующих в эксперименте. После того как все страны созданы, эти переменные должны быть равны. В начальный момент, когда еще не создана ни одна из стран, *countr=0*. Значение переменной *MaxCountr.js* отражает ограничение сценария имитационного эксперимента на максимальное количество участвующих в нем стран. Изменение этой переменной (особенно в бóльшую сторону) должно быть согласовано с реальными возможностями алгоритмов, реализующих работу макета.**

Далее в каталоге *Distr* находится переменная *Names.js*. В ней хранятся имена стран, участвующих в эксперименте, заданные в момент создания этих стран. Файл *Names.js* создает-

ся и изменяется автоматически скриптами модели и не предназначен для редактирования вручную.

Далее в каталоге *Distr* находятся переменные *TypesOfCountries.js* и *TypesOfDemogr.js*. В этих переменных реализована заложенная в сценарии макета возможность выбирать тип алгоритма вновь создаваемой страны и тип начальных данных для этой страны из трех заранее заданных типов (т. е. всего $3*3=9$ типов стран). **Изменение этих переменных должно быть согласовано с реальными возможностями алгоритмов, реализующих функционирование стран.**

Последняя переменная каталога *Distr* – *year.js* содержит значение текущего модельного года. Синтаксис задания года-синтаксис Java Script: ***var Gyear=1991;*** Имитационный эксперимент по сценарию начинается с 1991 г., поэтому начальное значение переменной ***Gyear – 1991.***

Для того чтобы начать имитационный эксперимент сначала, т. е. с создания стран заново, достаточно изменить две переменные каталога *Distr: countr.js*, назначив ее значение *var countr=0;*** и *year.js*, назначив ее значение ***var Gyear=1991;*** Эти значения можно изменить как непосредственно на информационном сервере, так и удаленно, например, через FTP, при наличии соответствующих прав доступа. Дальнейшее взаимодействие с моделью должно осуществляться только через интерфейс информационного сервера.**

Далее в каталоге *Distr* хранятся HTML-странички, представляющие главную страницу информационного сервера, а также страницу, предоставляющую возможность создания

стран. Это файлы: *index.htm*, *fr1.htm*, *fr2.htm*, *create.htm* а также относящиеся к ним рисунки: *game.gif* и *hrc.gif*.

При создании стран в каталоге ***Distr*** автоматически, соответствующим скриптом создается файл ***Model.inf***, являющийся текстовым файлом, каждая переменная которого занимает отдельную строку, предназначенный для агента доступа и содержащий следующую информацию о имитационном эксперименте:

<число стран, участвующих в эксперименте (1 - 4)>

<имя первой страны (строка)>

<тип алгоритма первой страны (0 – 2)>

<тип данных первой страны (0 – 2)>

<номер первой страны = 0>

. . .

<имя последней страны (строка)>

<тип алгоритма последней страны (0 – 2)>

<тип данных последней страны (0 – 2)>

<номер последней страны (0 – 3)>

Каталоги стран

В каталогах стран должны быть предустановлены лишь HTML-странички, отражающие состояние страны, и относящиеся к ним рисунки. Это файлы ***demogr.htm***, ***ecology.htm***, ***economy.htm***, ***fr1.htm***, ***index.htm***, ***test.htm*** и ***game.gif*** и ***hrc.gif***, одинаковые для всех стран. Все остальные переменные, рисунки и файлы создаются и изменяются автоматически в ходе создания и функционирования модели, и при нормальной работе не требуют ручного вмешательства администратора.

Из важных переменных отметим ***year.js***, она содержит текущий модельный год данной страны в формате: ***var year=1992;***. При нормальной работе модели значение переменной ***year*** равно значению переменной ***Gyear*** (***year.js*** из каталога ***Distr***), если страна еще не задала свои управления за текущий модельный год, или же ***Gyear+1***, если данная страна уже задала свои управления за текущий модельный год, а из оставшихся участников эксперимента не все успели это сделать.

Каталог CGI

Для работы скриптов CGI, на информационном сервере должен быть установлен интерпретатор языка Perl. В настоящее время используется свободно распространяемый ActivePerl версии 5.8.4.810 фирмы ActiveState Corp.

Все скрипты CGI, относящиеся к распределенной модели, расположены в подкаталоге ***Distr*** каталога ***cgi-bin***. Это скрипт ***test.cgi***, отвечающий за создание новых стран, и ***old.cgi***, отвечающий за обработку ежегодных управлений странами и выполнение очередного ежегодного модельного шага. Этими скриптами активизируются IAR-агенты доступа ***Init.exe*** и ***Step.exe*** соответственно, через командные файлы активизации соответственно ***Init.bat*** и ***Step.bat***. Для правильной работы агентам доступа необходим файл привязки к службе имен CORBA – ***ns.iior***.

Внимание! Данные агенты доступа не требуют ручной активизации. Они активизируются скриптами автоматически, в ходе функционирования макета.

Сообщения, создаваемые на этапе генерации стран и на каждом модельном шаге, записываются в файлы *init.log* и *step.log* соответственно.

Для единства стиля динамически создаваемых страничек, используются содержащиеся в каталоге рисунки *game.gif* и *hrc.gif*.

Файл *pass.pss* хранит в зашифрованном виде пароли доступа к странам, участвующим в эксперименте. **Этот файл создается автоматически скриптом и не предназначен для редактирования вручную.**

Распределенные ресурсы

К распределенным информационно-алгоритмическим ресурсам макета относятся алгоритмы функционирования экземпляров стран (в существующем варианте макета экземпляров может быть от 1 до 4), алгоритм функционирования мира в целом, база демографических данных. Каждый из этих ресурсов может быть реализован на отдельном удаленном компьютере.

Для удобства настройки этих ресурсов на удаленных компьютерах, все они объединены в один каталог **ACETAO**. Данный каталог должен быть переписан на удаленный компьютер, в переменную среды **PATH** этого компьютера должен быть включен путь к ...**ACETAO\bin**, после чего на компьютере может быть активизирован агент доступа к тому или иному ресурсу. На одном удаленном компьютере может быть активизирован один или несколько (до 6, 4 страны + мир + база

данных) агентов доступа к имеющимся информационно-алгоритмическим ресурсам.

Все информационно-алгоритмические ресурсы расположены в подкаталоге **IMITAT** каталога **АСЕТАО**. В частности, здесь расположен файл привязки к службе имен CORBA – **ns.ior**.

Для правильной работы распределенной модели необходимо, чтобы агенты доступа ко всем распределенным ресурсам, требующимся по сценарию имитационного эксперимента, были активизированы вручную (запущены соответствующие командные файлы в соответствующих каталогах удаленных компьютеров) до начала работы с моделью.

Ресурсы стран

Информационно-алгоритмические ресурсы стран расположены в подкаталогах каталога **IMITAT** с именами **country0, ..., country3** (максимальное число стран в макете в настоящее время – 4). Каждый из каталогов **countryX** содержит три подкаталога **TYPE_1, TYPE_2, TYPE_3**, в каждом из которых предустановлены (возможно, разные) алгоритмы создания и функционирования стран – исполняемые файлы с именами **ecdm_i_new.exe** и **Ecdm_new.exe** соответственно. Как уже упоминалось выше, при создании новой страны можно связать с ней один из этих алгоритмов.

Каталог страны **countryX** содержит также предустановленные программы создания и преобразования картинок, отображающих состояние страны на информационном сервере макета. Это программы **SocStr.exe** – генератор графиков со-

стояния страны после очередного года моделирования и ***bmp2png.exe*** – свободно распространяемая программа преобразования графиков из громоздкого формата битовых карт ***bmp*** в формат переносимой сетевой графики ***png***, более подходящий для представления графической информации в Интернете.

Далее каталог ***countryX*** содержит агент доступа к стране ***AA_Country.exe*** и командный файл активизации агента доступа ***Country.bat***.

Для правильной работы распределенной модели необходимо, чтобы каждый агент доступа к ресурсам всех стран был активизирован вручную (запущен командный файл *Country.bat*** в соответствующем каталоге) до начала работы с моделью.**

Все остальные файлы каталога ***countryX*** не являются предустановленными, они создаются и изменяются агентом доступа автоматически и не требуют администраторского вмешательства.

Ресурс «мир»

Информационно-алгоритмический ресурс «мир» расположен в подкаталоге ***WORLD*** каталога ***IMITAT***. Здесь заведены изначально пустые каталоги, соответствующие возможным странам – участницам имитационного эксперимента, с именами ***country0, ..., country***. Также в каталоге находится алгоритм, реализующий функционирование мира в целом – исполняемый файл ***World.exe***.

Агент доступа к этому ресурсу - исполняемый файл **AA_World.exe**, а командный файл его активизации - **World.bat**.

Для правильной работы распределенной модели необходимо, чтобы этот агент доступа был активизирован вручную (запущен командный файл *World.bat*) до начала работы с моделью.

Ресурс «база данных»

Информационно-алгоритмический ресурс «база данных» расположен в подкаталоге **DB** каталога **IMITAT**. Каталог **DB** содержит три подкаталога **DATA_1, DATA_2, DATA_3**, в которых хранятся три типа (возможно, различных) исходных демографических данных, каждый из которых можно связать с той или иной страной, в момент ее создания. Каталог **DB** содержит также агент доступа к данным - исполняемый файл **AA_DB.exe**, и командный файл его активизации - **DB.bat**.

Для правильной работы распределенной модели необходимо, чтобы этот агент доступа был активизирован вручную (запущен командный файл *DB.bat*) до начала работы с моделью.

Практическую работу по созданию системы моделей, ориентированных на анализ устойчивого социально-экономического развития России, по мнению авторов, следует начинать с компьютерной реализации модели демографического процесса в России в режиме распределенных вычислений. Конечной целью такой разработки является наличие в

Интернете в свободном доступе вычислительного процесса, воспроизводящего возрастную и половую структуру населения России по известным начальным распределениям (по умолчанию - данные переписи населения России 2002 г.), известным прогнозам коэффициентов рождаемости и смертности (по умолчанию - данные ЦЭМИ РАН) и известными возрастными и половыми распределениями мигрантов.

Этот вычислительный процесс оформляется как информационно-алгоритмический ресурс и доступен для использования с помощью технологии IARnet [4,13]. По умолчанию вычисляется «стандартный» набор характеристик демографического процесса как функций времени (количество детей, школьников, студентов, аспирантов, пенсионеров - мужчин и женщин, разность между количеством родившихся и умерших, среднее время жизни населения в целом, среднее время жизни женщин, среднее время жизни мужчин, среднее время жизни населения в целом, отношение количества активного населения к количеству пенсионеров и т.д.).

В то же время пользователю предоставляется возможность оформить как информационно-алгоритмический ресурс информацию, используемую вычислительным процессом по умолчанию: начальные данные, данные о коэффициентах рождаемости и смертности, данные о миграции. Кроме того, пользователь может оформить как информационно-алгоритмический ресурс характеристики демографического процесса, которые его интересуют и которые отсутствуют в списке характеристик, вычисляемых по умолчанию. После этого пользователь может запустить в режиме распределенных вычислений сконструированный им вычислительный процесс.

Реализация такого проекта при наличии подробной инструкции по оформлению информации и вычислительных процессов как информационно-алгоритмических ресурсов, а так-

же при наличии соответствующей службы, предоставляющей необходимые консультации, позволит любому пользователю, имеющему доступ в Интернет, получить интересующие его демографические характеристики. Несомненно практическое значение такого проекта, учитывая известную всем остроту демографической проблемы в России и широкий общественный интерес к этой проблеме.

Л и т е р а т у р а

1. Павловский Ю.Н., Белотелов Н.В., Бродский Ю.И. Имитационное моделирование. М.: Академия, 2008, 236 с.
2. Белотелов Н.В., Бродский Ю.И., Оленев Н.Н., Павловский Ю.Н. Эколого-социально-экономическая модель: гуманитарный и информационный аспекты //Информационное общество. № 6. 2001. С. 43-51.
3. Белотелов Н.В., Бродский Ю.И., Оленев Н.Н. и др. Проблема устойчивого развития: естественно-научный и гуманитарный анализ М.: Фазис, 2003. 96 с.
4. Форрестер Дж. Мировая динамика. М.: Наука. 1978. 165 с.
5. Капица С.П. Общая теория роста человечества. М.: Наука. 1999. 190 с.
6. Капица С.П. Глобальная демографическая революция и будущее человечества. //Новая и новейшая история, 2004. № 4.
7. Meadows D.L. et al. Dynamics of Growth in a Finite World. Cambridge, Mass. Whright Allen Press Inc.,1974.
8. Медоуз Д.Х., Медоуз Д.Л., Рандерс Й. За пределами роста. Изд. группа Прогресс, Пангея, 1994. 304 с.
9. Горшков В.Г. Физические и биологические основы устойчивости жизни. М.: ВИНТИ, 1995, 470 с.
10. Петров А.А., Поспелов И.Г., Шананин А.А. Опыт математического моделирования экономики. М.: Энергоатомиздат. 1996. 544 с.
11. Мусеев Н.Н. Агония России: Есть ли у нее будущее. Попытка системного анализа проблемы выбора. Зеленый мир. 1996. № 12. С. 6-15.

12. *Ташлицкая Я.М., Шананин А.А.* Многоукладность технологической структуры и влияние транзакционных издержек на распространение инноваций. //Математическое моделирование. 2000. Т.12, №12. С. 24-34.
13. <http://www.globalinsight.com/>
14. *Моисеев Н.Н., Павловский Ю.Н.* Информатика как отрасль народного хозяйства: особенности, темпы и тенденции развития// Экономика и математические методы. 1986. № 5.
15. *Павловский Ю.Н.* Имитационные модели и системы. М.: Фазис. 2000. 131с.
16. *Бродский Ю.И., Лебедев В.Ю.* Инструментальная система имитации MISS. М.: ВЦ АН СССР, 1991. 180 с.
17. *Афанасьев А.П., Волошинов В.В., Кривцов В.Е.* О возможных принципах организации доступа к удаленным вычислительным ресурсам на основе CORBA. Тез. докл. 1-й Московской конференции «Декомпозиционные методы в математическом моделировании». М.: ВЦ РАН, 2001. С. 3-4.
18. *Афанасьев А.П., Волошинов В.В., Кривцов В.Е. и др.* Использование информационно-алгоритмических ресурсов для организации распределенных вычислений. //Проблемы вычислений в распределенной среде: Сб. тр. ИСА РАН, М.: Эдиториал УРСС, 2004.
19. *Афанасьев А.П., Волошинов В.В., Рогов С.В. и др.* Развитие концепции распределенных вычислительных сред. //Проблемы вычислений в распределенной среде: организация вычислений в глобальных сетях. Труды ИСА РАН. М.: РОХОС, 2004. С. 6-105.

20. *Popkov Yu.S.* Dynamic models of macrosystems with self-reproduction and their application to the analysis of regional systems, 1993. The Annals of Regional Sciences, 27.
21. *Моисеев Н.Н.* Идеи естествознания в гуманитарной науке: о единстве естественнонаучного и гуманитарного знания //Человек. 1992. Вып. 2. С. 5-16.
22. *Павловский Ю.Н.* Математический и гуманитарный анализ механизма ядерного сдерживания //Вестник РАН. 2000. № 4. С. 195-202.
23. *Моисеев Н.Н., Александров В.В., Тарко А.М.* Человек и биосфера. Опыт системного анализа и экспериментов с моделями. М.: Наука. 1985. 385 с.
24. *Павловский Ю.Н.* Экологический контроль — составная часть индустрии //Вестник РАН. 1993. № 2, С. 29-38.
25. *Староверов О.В.* Азы математической демографии. М.: Наука, 1997. 159 с.
26. *Леонтьев В.В.* Экономические эссе. Теория, исследования, факты и политика: Пер. с англ./ Под. ред. С.С. Шаталина, Д.В. Волового М.: Политическая литература. 1990. 415 с.
27. *Бродский Ю.И.* Эколого-социально-экономическая имитационная модель: технология реализации. //Моделирование, декомпозиция и оптимизация сложных динамических процессов. М.: ВЦ РАН, 2001. С. 89-107.
28. *Бродский Ю.И.* Декомпозиционные методы в инструментальных средствах имитации. Тез. докл. 1-й Московской конференции «Декомпозиционные методы в математическом моделировании». М.: ВЦ РАН, 2001. С. 21-22.
29. *Бродский Ю.И.* Проблемы создания центра имитационного моделирования в ИНТЕРНЕТ. //Моделирование, де-

композиция и оптимизация сложных динамических процессов. М.: ВЦ РАН, 1998. С. 29-35.

30. *Бродский Ю.И.* Разработка инструментальных средств имитационного моделирования: сложных технических комплексов для параллельных вычислительных систем. //Моделирование, декомпозиция и оптимизация сложных динамических процессов. М.: ВЦ РАН, 1999. С. 33-92.

Оглавление

Введение	3
1. Эколога – демографо - экономическая модель	8
Описание базовой модели	8
Модель демографического процесса	8
Виды экономической деятельности, учитываемые в эколого- социально-экономической имитационной модели	10
Технологии в ЭДЭМ	12
Экономические агенты в ЭДЭМ	13
Модель производственных процессов	14
Модель процесса загрязнения окружающей среды	18
2. Проект распределенной системы прогнозирования демографических процессов в системе регионов	20
Модель демографического процесса нижнего уровня	20
Система социально-образовательно-профессиональных характеристик населения регионов	23
О понятии «когорты»	26
3. Концепция построения инструментальной системы распределенного моделирования	37
Компонента - основная составляющая распределенной модели	38
Реакция компоненты на события	39
Анализ метамоделей (комплекса)	40
Синтез метамоделей (комплекса)	42

Общая архитектура инструментальной системы распределенного имитационного моделирования	47
Как описываются комплексы и компоненты.....	50
Язык описания комплексов и компонент	50
Определение типов записей фазовых переменных и констант	51
Операторы коммутации	55
Синтаксис описателей комплексов	57
Синтаксис описателей методов.....	59
Синтаксис описателей компонент.....	60
Компиляция описателей компонент модели.....	63
Компоновка модели	70
Работа модели во время эксперимента	71
Работа с моделью во время имитационного эксперимента	72
4. Компьютерная реализация макета распределенной эколого- демографо-экономической модели в режиме распределенных вычислений	72
Описание агентов доступа	78
Интерфейс модели	81
Организация имитационного эксперимента	94
Подготовка распределенного имитационного эксперимента	95
Информационный сервер модели	95
Каталог Distr	95
Каталоги стран	98

Каталог CGI	99
Распределенные ресурсы.....	100
Ресурсы стран	101
Ресурс «мир»	102
Ресурс «база данных»	103
Л и т е р а т у р а	106