

## КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЕСТЕСТВЕННЫХ И СОЦИАЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СПЕЦИАЛЬНОГО ПРОФИЛЯ UML\*

Гурьянов В.И., e-mail: vg2007sns@rambler.ru

Визуальные модели широко используются в существующих технологиях проектирования программного обеспечения (ПО). Естественно ожидать расширение сферы применения визуального моделирования в системах имитационного моделирования. Визуальное моделирование, в частности, может быть реализовано в специальном UML–профиле, предназначенном для компьютерного моделирования систем различной природы. В данном докладе предложен один из проектов такого профиля, далее называемый как *научный профиль (Scientific Profile)*.

Назначение профиля определим как средство формализованного описания компьютерных моделей (далее преимущественно обсуждаются имитационные модели). Р.Шеннон в своей книге [1] высказал утверждение, сохранившее актуальность и по сей день, сказав, что создание имитационной модели в большей мере, чем программирование является искусством. За время, прошедшее с момента выхода книги, программирование стало инженерией, а ситуация с имитационным моделированием по существу осталась прежней. Как будет показано далее, *Scientific Profile* предоставляет возможность создания повторяемого, предсказуемого и контролируемого процесса разработки имитационных моделей.

Концептуальная модель профиля (профиль UML – это набор стереотипов, помеченных значений и ограничений) кратко может быть представлена следующим образом. Предполагается, что модель предметной области представлена в эксплицитной форме, например, как онтология. Профиль устанавливает множество отношений между концептами и стереотипами профиля (стереотип может определять новые метки и ограничения, которые не являлись частью исходного элемента). В профиль вводится *дуальная семантика* посредством механизма помеченных значений для стереотипов в виде {Concept = Имя концепта}. Для разграничения семантик, можно говорить о *вычислительной семантике* (см., например, [2]) и о *предметной семантике* стереотипа. Профиль, будучи расширением UML, может использоваться в разнообразных методологиях проектирования ПО (Unified Process [3], порождающее программирование и др.). Идея предлагаемого подхода состоит в

том, чтобы перенести эти методологии на разработку имитирующих программ. Вследствие дуальной семантики, таким образом, можно будет говорить о методологиях построения компьютерных (имитационных) моделей.

Онтологии подразделяются на онтологии верхнего уровня, онтологии предметной области, онтологии задач и прикладные онтологии. Поэтому каждый элемент модели может иметь несколько концептов. Стереотипы субпрофилей (например, *Scientific Profile for Classical Mechanics*) находятся в отношении обобщения со стереотипами *Scientific Profile*. Далее будем полагать, что один концепт определен в *Scientific Profile*, другой - непосредственно в онтологии предметной области (см. рис.1; уровень субпрофиля опущен).

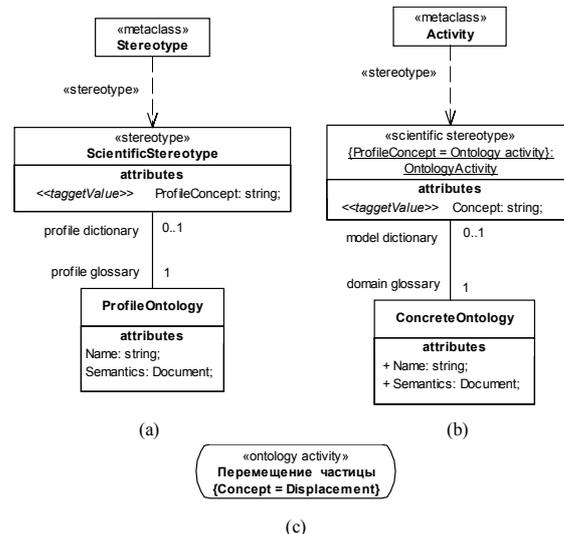


Рис.1. Пример определения дуальной семантики: (а) метамодель профиля, (б) определение стереотипа профиля, (с) применение стереотипа

UML является формальным языком, теория которого может быть построена с использованием RM-ODP (The Reference Model of Open Distributed Processing). Мы будем рассматривать *Scientific Profile* с позиций экстенционального подхода к определению семантики языка, где в качестве универсума экстенционалов будет выступать онтология предметной области.

\* Здесь представлена исправленная и дополненная версия доклада; см. Гурьянов В.И. Профиль UML для имитационного моделирования социально-экономических систем // XVIII Международная конференция "Математика. Экономика. Образование". Тезисы докладов. – Изд-во СКНЦ ВШ ЮФУ, Ростов н/Д, 2010. – С. 165-166; публикация, см. Гурьянов В.И. Специальный UML–профиль для моделирования сложных систем // Информационные технологии моделирования и управления. – Воронеж: Изд-во «Научная книга», 2010. – № 3(62). – С. 356-362. (см. http://www.sbook.ru/itmu/itmubook.htm)

Приведем неформальное определение некоторых ключевых стереотипов и их концептов уровня *Scientific Profile*. Далее всюду термины профиля выделены наклонным шрифтом.

Если рассматривать код имитирующей программы как текст, описывающий предметную область, то UML-профиль будет играть роль метаязыка. Прагматику зададим стереотипом *Research Use Case Model* (расширяет "metaClass" Use Case Model), семантику - *Research Analysis Model* ("metaClass" Analysis Model), а синтактику - *Research Design Model* ("metaClass" Design Model).

Стереотип *Research Use Case Model* определяет цели и намерения *Исследователя* (стереотип *Researcher* "metaClass" Actor, {ProfileConcept = Researcher}), последние обслуживаются прецедентами (*Приготовить начальное состояние*, *Вычислить новое состояние*, *Обработать результаты наблюдения* и др.). Среди ролей *Исследователя* отметим роли *Наблюдатель контекста* и *Наблюдатель системы*, и будем различать их тем, где расположены датчики измерений (прежде всего детекторы и хронометры). Пример диаграммы прецедентов приведен на рис.2.

*Research Analysis Model*, представленная пакетом *Analysis System* со стереотипом *Универсум* (<<Universe>> {ProfileConcept = Universe}), с точки зрения вычислительной семантики представляет собой каркас. Архитектура *Research Analysis Model* определяется четырьмя пакетами, распределенными по двум уровням абстракции (специфическому и общему) и двум разделам (*гносеологическому* и *онтологическому*); см. пример на рис.3.

На специфическом уровне заданы пакеты со стереотипами *Исследовательская установка* (<<Research Instruments>>, {ProfileConcept = Research instruments}) и *Мир* (<<World>>, {ProfileConcept = World}). Вычислительная семантика этих пакетов может быть задана паттернами проектирования, например, MCV и Composite соответственно. Стереотип *Измерение* (<<Measurement>> "metaClass" <<permit>>, {ProfileConcept = Measurement}) применяется к отношению зависимости, связывающее данные пакеты. Стереотип *World* отражает исследуемую систему и ее окружение и содержит описание имитационной модели в традиционном понимании.

Пакеты со стереотипами *Epistemology Entity* ({ProfileConcept = Epistemology}) и *Ontology Entity* ({ProfileConcept = Ontology}) общего уровня задают методологии проведения измерений и законы функционирования системы в стереотипах UML. Их вычислительная семантика определяется как библиотеки базовых классов. В пакетах онтологического раздела, запрещено использовать какие либо классы, кроме помеченных меткой *Concept*. В корневого пакет *TopLevel* заносится класс *Magnitude* и все или некоторые из его потомков. Это обеспечивает доступ всех пакетов к данному классу. Предметная семантика этих классов определяется, как агент взаимодействия гносеологических и онтологических объектов и предназначена для определения

общего типа интерфейса. Выбор этих классов определяет набор доступных измерительных шкал (классификационные, сравнительные, количественные).

Изучаемые или *онтологические сущности* (<<Ontology Entity>>) задаются стереотипами *Окружающая среда* (<<Ontology Environment>> "metaClass" Class), *Система* (<<Ontology System>> "metaClass" Class), *Атом* (<<Ontology Atom>> "metaClass" Class) и *Активность* (<<Ontology Activity>>).

Центральное место в *Scientific Profile* занимает стереотип *Активность* (тип - Behavioral things, "metaClass" Activity стандарта UML, см. рис.1), который фактически предопределяет семантику остальных стереотипов. В различных областях науки отмечены факты, когда структуры и системы имеют разные темпы развития. Для обозначения этого феномена иногда используют понятие "темпомир" (см. например, [4]). Предельная, абсолютная форма этого понятия составляет предметную семантику стереотипа *Ontology Activity*, а концепт можно назвать *абсолютным темпомиром*. С точки зрения *наблюдателя контекста* *абсолютный темпомир* представляет собой событие, не имеющее продолжительности. С точки зрения *наблюдателя темпомира*, контекст *абсолютного темпомира* представляет собой статическую картину, а изменение доступных переменных контекста запрещено. Под *единичным квантом существования* системы будем понимать атомарную (неразложимую) активность, наблюдаемую *наблюдателем контекста*, т.е. действие (action). С точки зрения *наблюдателя темпомира*, *абсолютный темпомир* определяется как сеть событий и причинно-следственных связей и описывается ориентированным графом (точнее – нагруженным мульти- псевдо- гиперграфом), представлением которого в UML является диаграмма деятельности (Activity diagram) или диаграмма состояний (State diagram). *Время* – отображение конечного подмножества целых чисел на множество вершин графа. Это отображение выполняет специальный прибор (гносеологическая сущность) – счетчик времени, который следует отличать от хронометра. Тем самым, в *Scientific Profile* время не является онтологической сущностью, а представляет собой данные измерений. Под *пространством* будем понимать объект со стереотипом *Ontology Space*; предметная семантика определяется как объект для размещения других объектов, образующих структуру системы. Определяющий признак *пространства* – два и более объекта не могут быть в одной и той же ячейке пространства. Поэтому *пространство процесса* – вся совокупность переменных, доступных процессу. Вычислительная семантика <<Ontology Space>> близка к понятию "структура данных".

*Абсолютный темпомир* не является объектом, хотя и может быть инкапсулирован в объекте. В языках программирования (вычислительная семантика) *абсолютный темпомир* описывается процедурой; создание *абсолютного темпомира* – это вызов процедуры из текущей процедуры. Абстракция темпомира позволяет представить природный процесс как совокупность вложенных, параллельных/последовательных, синхронизированных/асинхронных и открытых/замкнутых темпомиров, и, тем самым, снимает многие логические проблемы непротиворечивого описания.

Примером может служить проблема изотаксии для физического дискретного пространства в классической механике.

Предметную семантику стереотипа *Ontology Environment* определим посредством дихотомии понятия "мир" относительно понятия "объект изучения", т.е. весь *мир*, кроме изучаемого объекта. С точки зрения системного анализа объекты с этим стереотипом определяют функцию системы относительно окружающей среды и позволяют задать ее системные характеристики (т.е. ее системную динамику). Окружающая среда задается крайне схематично, однако ее связи с системой определяются с необходимой точностью. Как правило, анализируются горизонтальные и вертикальные связи. Предметная семантика стереотипа *Ontology Atom* определяется как конечный уровень декомпозиции изучаемой системы на подсистемы (в частности, это могут быть акторы и программные агенты). Композиция подобных объектов моделирует подсистему также весьма схематично, но их внешние связи определяются с необходимой точностью. *Ontology System* является инвариантом относительно выбора *Исследователя*; *Ontology Environment u Ontology Atom*, напротив, определяются положением *Исследователя* (в связи с этим отметим идеи, развиваемые в эндофизике). Для описания взаимодействий объектов может быть использована диаграмма кооперации; см. рис.4.

*Research Design Model* определяет правила (способ) описания модели на конкретном языке программирования и позволяет отделить элементы имитационной модели от ограничений, налагаемых синтаксисом этих языков (в данной работе всюду предполагается использование Smalltalk, что, конечно же, не исключает GPSS World, AnyLogic и др.). Основной проблемой потока работ по созданию проектной модели является представление параллельных процессов, определенных в модели анализа, в виде последовательных операций, пригодных для описания на выбранном языке программирования. Затем вычислительный алгоритм должен быть исследован на точность и устойчивость.

Конечный результат изучения симуляции (артефакт имитационного моделирования) – дерево сценариев с указанием количественных значений критических характеристик. Сценарии функционирования, развития или эволюции систем могут быть представлены в форме диаграмм последовательности действий; см. пример на рис.5.

Приведем несколько характерных примеров применения профиля, выбрав их таким образом, чтобы предметные области по возможности существенно отличались друг от друга. Объектные модели будем описывать в порядке следования рабочих потоков (определение требований, анализ, проектирование, реализация и тестирование) Unified Process, см. [3].

Пример 1. Механический осциллятор. Для симуляции механического движения рассмотрим перемещение объекта по ячейкам дискретного пространства [5].

*Research Use Case Model* является типичной и содержит прецеденты,

описанные выше (см. рис.2). *Research Analysis Model* содержит пакеты *SI* (*<<epistemology entity>>*, {Concept = *Système International d'Unités*}), *Registrar* (*<<research instruments >>*, {Concept = Регистратор механического движения}), *Cosmos* (*<<world>>*, {Concept = Вселенная Ньютона}) и *Classical mechanics* (*<<ontology entity>>*, {Concept = Классическая механика}). Пакет *Classical mechanics* определяет законы механического движения и представлен классами *Substance*, *MechanicalSystem*, *Particle* и четырьмя активностями со стереотипом *<<ontology activity>>*: *Displacement* ({Concept = Перемещение}), *Interaction* ({Concept = Взаимодействие}), *Collision* ({Concept = Столкновение}) и *Disintegration* ({Concept = Распад}). Постулируется, что *Перемещение* и *Взаимодействие* образуют параллельные активности, а *Столкновение* – активность, порожаемая активностью *Перемещение*. *Распад* – активность, инкапсулированная в объекте класса *MechanicalSystem*. Все классы являются потомками одного абстрактного класса *Substance*. Композиция этого класса содержит статическую структуру *space* (*<<ontology space>>*, {Concept = Физическое пространство}), которая моделирует 1-мерное пространство, наделенное топологией и метрикой. Пакет *Cosmos* содержит реализации прецедентов "Приготовить начальное состояние", "Вычислить следующее состояние" и классы *Cosmos*, *Oscillator* и *Field*. Объект класса *Cosmos* моделирует контекст объекта *oscillator*, содержит датчики для измерения амплитуды и периода колебаний и не является космологической моделью. Объект класса *Field* схематически моделирует взаимодействие материальных точек и не является моделью физического поля.

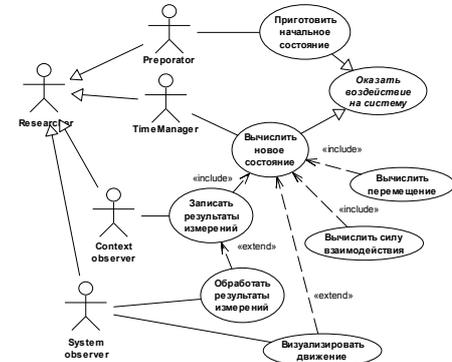


Рис.2. Диаграмма прецедентов для симуляции механического движения

*Research Design Model* задает аппроксимацию параллельных активностей *Displacement* и *Interaction* в виде последовательных деятельностей и в простейшем случае описывается модифицированным методом Эйлера.

Конечный артефакт имитационного моделирования – сценарий функционирования системы, - задается единственной диаграммой последовательности действий, которая в нашем случае может быть получена аналитическим путем из диаграммы кооперации.

Пример 2. Экономика и анализ проблемных ситуаций. В таких областях, как психология, социология, экономика и т.п., где применение традиционного математического аппарата вызывает известные сложности, можно ожидать наибольшего эффекта от применения *Scientific Profile*. Это ожидание можно объяснить тем, что классы можно использовать как фреймы (рамки) посредством которых можно структурировать сложные системы.

В качестве типичного примера разработки имитационной модели в экономике рассмотрим макроэкономическую систему, например, социально-экономическую систему (СЭС) региона. За основу модели примем обобщенную макроэкономическую модель Дж.М.Кейнса [7]. Далее мы будем предполагать, что модель Кейнса представлена онтологией. Пусть до некоторого момента времени  $T$  экономика находилась в состоянии равновесия, т.е. совокупный спрос равен предложению:  $Y_D(t) = Y_S(t)$ . Допустим, что в момент  $T$  совокупный спрос изменится. Цель *Исследователя* – определить новое равновесное состояние. Отметим, что это не динамическая (как в Примере 1), а квазистационарная задача.

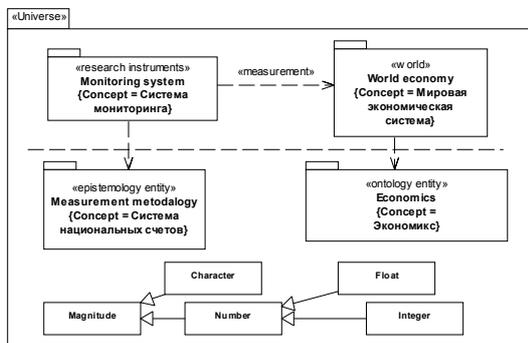


Рис.3. Архитектура пакета Analysis System для модели региона

*Research Analysis Model*, представленная пакетом Analysis System, показана на рис. 3. Пакет Economics определяет законы функционирования

СЭС и представлен классами Subject, ComplexSubject, Person и четырьмя активностями со стереотипом `<<ontology activity>>`, моделирующими фазы общественного воспроизводства: *Production* ({Concept = Производство}), *Distribution* ({Concept = Распределение}), *Exchange* ({Concept = Обмен}) и *Consumption* ({Concept = Потребление}). Постулируется, что все активности являются параллельными.

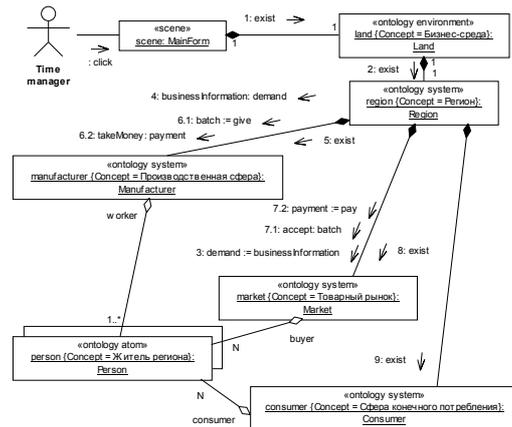


Рис. 4. Диаграмма кооперации, моделирующая субъектно-объектную структуру общественного воспроизводства

Все классы являются потомками одного абстрактного класса Subject. Композиция этого класса содержит структуру space (`<<ontology space>>`, {Concept = Структура субъекта}). Пакет World Economy содержит реализации прецедентов "Приготовить начальное состояние", "Вычислить следующее состояние" и классы Land, Region, Manufacturer, Market и Consumer. Одноименные экземпляры классов используются для создания симуляции. Изучаемую систему моделирует объект region, контекст которой задается объектом land. Композиция объекта region включает объекты, моделирующие основные подсистемы: manufacturer (сфера производства), market, consumer (сфера потребления). Атомарный объект person моделирует жителя региона. Деньги, документы и продукты производства, в целях упрощения, моделируются объектами, имеющими класс Integer. Кооперация объектов, реализующих прецедент "Вычислить следующее состояние" показана на рис. 4 (Для модели региона также следует рассмотреть *Правительство* и *Банк*,

которые здесь не показаны). Каналом передачи сообщений является внутренняя среда объекта region.

Research Design Model определяет аппроксимацию параллельных активностей Production, Distribution, Exchange и Consumption в виде последовательных деятельностей. Для анализа устойчивости удобно рассмотреть итерационное соотношение для  $Y_S$  и ввести новую переменную  $y_t = Y_S(t) - Y_E$ , которая равна отклонению текущего значения национального дохода от его нового равновесного значения  $Y_E$ . Если производственная функция и функция спроса линейны, поведение этой переменной описывается формулой геометрической прогрессии:  $y_{t+1} = cy_t$ , где  $0 < c < 1$ , т.е. в этом случае имеем бесконечно убывающую последовательность, модуль которой стремится к нулю. В виду линейности соотношения подобным же образом ведет себя начальная ошибка  $\delta y_t$ , т.е. алгоритм устойчив. Однако, как показано в монографии [7], в случае нелинейности возможно возникновение детерминированного хаоса, а алгоритм может стать неустойчивым.

Конечным артефактом имитационного моделирования будет дерево сценариев, представленное совокупностью вложенных пакетов диаграмм последовательности действий. Как правило, аналитическое решение можно получить на основе анализа диаграммы кооперации (рис.4) только для некоторых случаев (см. рис.5).

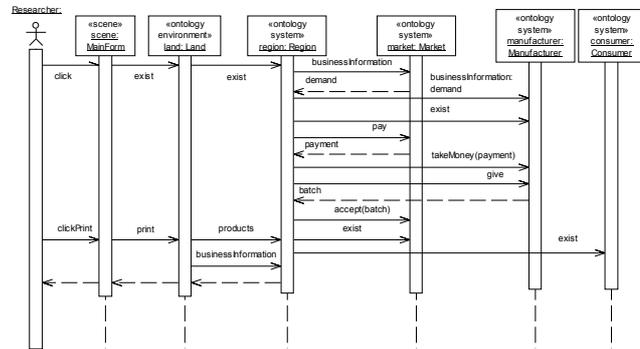


Рис. 5. Диаграмма последовательности действий, представляющая сценарий функционирования социально-экономической системы

Необходимость выйти за пределы традиционного системного анализа наиболее наглядно видна в анализе проблемных ситуаций. Рассмотрим в качестве примера задачу о двух заместителях. Пусть поток документов обрабатывается двумя заместителями. Документ должен быть утвержден

обоими заместителями прежде, чем он будет утвержден руководителем организации. Каждый заместитель имеет собственные критерии оценки документа и поэтому в некоторых случаях их мнения по одному и тому же документу могут не совпадать. Окончательное решение принимает руководитель.

Проблема, связанная с моделированием проблемных ситуаций, состоит в том, что возникают трудности с корректным применением стереотипа World. С одной стороны, руководитель должен получить сообщение от своих заместителей в форме документа. С другой стороны, каждый заместитель должен оценить документ согласно своему контексту ("картине мира", в терминологии Чекленда [8]), а если задан контекст, то определен мир, и вне мира уже ничего наблюдать не может. Последнее надо понимать как запрет на обмен сообщениями между объектами, моделирующими миры, и различать понятия "ненаблюдаемый объект" и "ничто". Разрешение данной логической трудности, на наш взгляд, возможно на основе включения в онтологию уровня профиля модальной логики Сола Крипке [9].

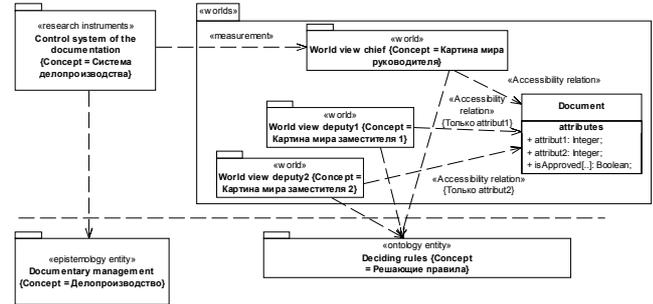


Рис.6. Архитектура пакета Analysis System (Research Analysis Model) для задачи двух заместителей

В логике Крипке вводится понятие *модельной структуры*  $\langle G, K, R \rangle$ , в которой  $K$  является множеством возможных миров,  $G$  является *актуальным* (действительным) *миром* —  $G \in K$ , а  $R$  является отношением между мирами. Актуальный мир будем определять как мир, в котором находится *Исследователь*. Трансмировые связи опишем стереотипом *Отношение Достижимости* ( $\langle \langle \text{Accessability Relation} \rangle \rangle$ , "metaClass" Dependency, {ProfileConcept = Accessability relation}), см. рис. 6.

Вычислительную семантику стереотипа *Accessability Relation* определим как отношение зависимости, обусловленное *разделяемыми переменными* (в терминологии

Smalltalk). Переменные, доступные более чем одному объекту, объединяются в группы, называемые *пулами*. Класс имеет два или более пулов, доступных его экземплярам. Один пул, содержащий глобальные переменные (global variable), доступен всем объектам системы. Каждый класс может иметь также второй пул переменных (class variable), который содержит переменные класса и доступен только его экземплярам. Классу могут быть доступны специально созданные пулы переменных (pool variable), которые доступны нескольким классам. Достижимость между мирами, в нашем случае, может быть описана, в частности, посредством переменных типа pool variable. Как видно из сравнения рис. 3 и 6, использование *пакета миров* (стереотип *Worlds*) вместо одного *мира* практически ничего не меняет в архитектуре модели анализа. Актуальный мир, представляющий собой все сущности пакета Analysis System кроме *пакета миров*, также может рассматриваться, в описанном выше смысле, как один из *миров*.

Введение *пакета миров* (или пакета "картин мира") позволяет применить методологию "мягких" систем Чекленда [8], которая включает семь этапов, для организации плана вычислительного эксперимента. Как легко видеть задача о двух заместителях решается в два прогона этого семеричного цикла. Хотелось бы подчеркнуть, что введение логики Крипке в онтологию уровня профиля связано с внутренней структурой профиля, а не только с задачами анализа проблемных ситуаций. Как показывает опыт построения объектных моделей, "жесткие" системы скорее исключение, чем правило.

Кратко остановимся на анализе бизнес-процессов. Это необходимо хотя бы потому, что именно Ericsson-Penker Profile [10] послужил отправной точкой по разработке *Scientific Profile*. Ряд современных методов моделирования бизнес-процессов основан на использовании языка UML. Среди таких методов наиболее известными являются метод Ericsson-Penker и метод, реализованный в технологии Rational Unified Process (RUP). Общим в *Scientific Profile* и в Ericsson-Penker profile является то, что в обоих случаях используется так называемый процессный подход к моделированию бизнес-процессов. Его основной принцип заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Общим в *Scientific Profile* и в профиле RUP Business Modeling является использование нескольких моделей (методика моделирования RUP предусматривает построение двух моделей: Business Use Case Model и Business Analysis Model). Основным отличием *Scientific Profile* от обоих профилей является то, что в нашем случае объект исследования моделируется в контексте познавательной ситуации. Бизнес-среда определяется в пакете со стереотипом *World* как онтологический контекст.

Пример 3. Моделирование ментальных сущностей. *Scientific Profile* применим не только для моделирования сущностей физического мира, но и для моделирования сущностей субъективной реальности, а также других программных сущностей (например, симуляция симуляции). Речь идет о психологии, лингвистике, математике и т.п. Например, искусство

демонстрирует поразительную аналогию между актом научного познания и восприятием художественного образа. Особенность объектных моделей художественных произведений состоит в том, что акторы моделей, как правило, лишены какой-либо ментальности. Их поведение определяется исключительно сюжетом (переменная типа class variable). Шерлок Холмс и доктор Уотсон стреляют в собаку Баскервиль не потому, что они увидели ее, а потому, что наступило следующее сюжетное событие. *Активности* инкапсулированы в объектах (хронотопах, в терминологии искусствознания), а реализация сюжета описывается паттерном Strategy.

Особый интерес представляет объектное моделирование математических объектов. Благодаря аксиоматической структуре математических теорий соответствующие субпрофили могут быть построены в обозримом виде и, далее, могут использоваться как шаблоны для построения субпрофилей в других областях знаний. В качестве иллюстрации приведем объектную модель дискретной группы вращения равностороннего треугольника.

*Research Use Case Model* определяется четырьмя ролями *Researcher* (по одному для каждого прецедента) – конфигурировать систему, вычислить алгебраическое выражение, документировать результат вычисления, документировать процесс вычисления.

*Research Analysis Model* содержит объекты `aAlgebraicExpression` (онтологический контекст; определяет процесс вычисления алгебраического выражения и в общем случае определяется через понятие "математический дискурс" [11]), `aElementA`, `aElementB`, `aElementI` (элементы группы  $a, b, I$ , где  $aa^{-1}=I$ ), датчик результата и датчик операций. Классы всех объектов являются потомками абстрактного класса `AlgebraicObject`, который имеет метод с селектором `multiplyBy: aAlgebraicObject` (см. рис.7). Элементы группы различаются контроллерами умножения, каждый из которых реализует соответствующую строку таблицы умножения. Элементы группы `aElementA` и `aElementI` являются атомарными объектами. Элемент `aElementB` представляет собой составной элемент, имеет класс `ComplexElement` и выражается через образующий элемент  $b = a^2$ . Кооперация объектов задается паттерном поведения Strategy.

По приведенной схеме могут быть построены многие математические объекты. В частности, это верно для многочленов, которые, как известно, являются одним из основных математических объектов компьютерной алгебры. Системы компьютерной алгебры, даже если используется объектно-ориентированное программирование, основаны на символьных вычислениях. Вычисление с объектами не является символическими вычислениями, а являются обобщением численных методов на объекты нечисловой природы. Формально методы вычислений (или, как говорят, компьютеринга) объектов могут быть описаны на языке аппликативных вычислений [12].

Для модели характерно рассмотрение элементов, без каких либо свойств (единственно, что можно сказать, так это то, что они принадлежат разным

классам), уникальность которых определяется исключительно способом их взаимодействия друг с другом. Т.к. обмен сообщениями – это связь элементов, то описанный выше подход выражает философию структуриализма (см. обзорную часть [11]). Заметим, что альтернативный подход – определение математических элементов с набором свойств, - приводит к компьютерной алгебре.

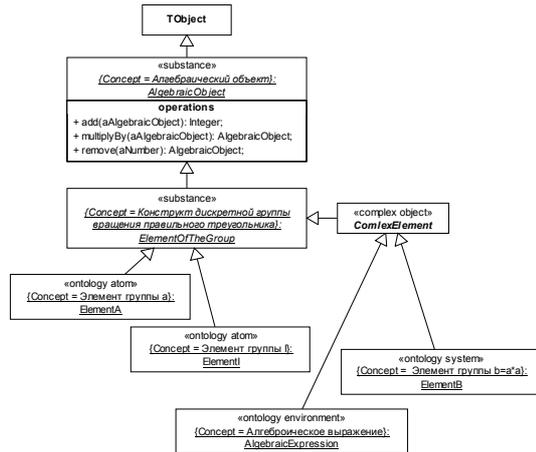


Рис.7. Диаграмма классов для моделирования дискретной группы вращения правильного треугольника

Предметная область математики лежит в идеальной сфере. Математика имеет дело с абстрактными объектами. Такие объекты обладают некоторым набором отношений, по которому их можно сопоставить с объектами физического мира (и тем самым определить математическую модель). Для действительного мира модель будет компьютерной, а непосредственно для ментальной сферы - имитационной.

Итак, профиль является визуальным языком описания объектных моделей. В данном докладе продемонстрированы возможности объектного моделирования систем различной природы посредством *Scientific Profile*. Профиль органически объединяет агентное, дискретно-событийное моделирование и системную динамику в единой модели. Основным качеством

профиля следует считать возможность применения формализованных методов в процессе построения имитационных моделей систем.

Дуальная семантика определяет двунаправленные связи между концептами и элементами UML. Выше мы интерпретировали элементы UML в понятиях предметной области (программирование как моделирование). Однако верно и то, что понятия предметной области могут быть описаны в понятиях вычислительной семантики (моделирование как программирование). Поэтому другим результатом применения профиля, возможно не менее значимым, мы считаем возможность представить естественные, общественные и экономические модели в абстрациях вычислений, объектов и сообщений, что хорошо согласуется с парадигмой компьютерных теорий (Computational economics, sociology) (см., например, [13]). Вполне естественным представляется применение *Scientific Profile* в системном анализе.

## Литература

1. Шеннон Р. Имитационное моделирование систем. - Искусство и наука. - М.: Мир, 1978. - 417 с.
2. Лавров С., Программирование. Математические основы, средства, теория. - БХВ-Петербург, 2001. - 314 с.
3. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. - Пер. с англ. - СПб: Символ-Плюс, 2007. - 624 с.,
4. Князева Е.Н., Курдюмов С.П. Основания синергетики. Режимы с обострением, самоорганизация, темпомиры. - СПб: Алетейя, 2002. - 414 с.
5. Вьяльев А. Н. Дискретное пространство-время. — М.: КомКнига, 2007. - 400 с.
6. Лебедев В.В. Математическое моделирование социально-экономических процессов, М.: Изограф, 1997. - 224 с.
8. Checkland P.B., Scholes I. Soft Systems Methodology in Action. Chichester: Wiley, 1990.
9. Крипке СА Семантическое рассмотрение модальной логики // Семантика модальных и интенциональных логик М., 1981; Крипке СА. Теорема полноты в модальной логике // Фейс Р. Модальная логика. М., 1974.
10. Eriksson H., Penker M. Business Modeling with UML: Business Patterns at Work. - John Wiley & Sons, 2000. - 459 p.
11. Гутнер Г.Б., Онтология математического дискурса. Структура и сущность в математическом рассуждении. М.: Издательство Московского Культурологического Лицея, 1999.
12. Вольфенгаген В.Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. М.: JurInfoR Ltd., АО "ЦентрЮриИнфоР", 2004.- 789 с.
13. Давыдов А.А. Компьютационная теория социальных систем // Социологические исследования. 2005. - № 6. С. 14-24.