

3.4. Порівнюються четверті параметри лінгвістичних складових A_1 та A_2 . На основі порівняння визначається відсоток збігу $\eta(A)$.

Визначення відсотку збігів за різними компонентами логіко-лінгвістичної моделі електронного текстового документу дає можливість проаналізувати характер збігів, а також визначити засоби утворення текстових дублікатів.

Література

1. Вавіленкова А.І. Застосування формальних алгоритмів у структурній лінгвістиці /А.І. Вавіленкова // Вісник Національного університету «Львівська політехніка». – 2014. – № 800. – С. 265–272.
2. Филиппов К.А. Лингвистика текста. Курс лекций. – Спб.: Издательство С.-Петербургского университета, 2008. – 336с.
3. Головкина С.Х. Лингвистический анализ текста //С.Х. Головкина, С.Н. Смольников. – Вологда: Издательский центр ВИРО, 2006. – 124с.
4. Вавіленкова А.І. Методологічні основи автоматичного аналізу логіко-лінгвістичних моделей текстових документів /А.І. Вавіленкова // Математичні машини та системи. – 2015. – № 1. – С. 65–71.

УДК 004.94

БАНК ТЕСТОВЫХ МОДЕЛЕЙ ДЛЯ ЯЗЫКА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ UML SP

В.И.Гурьянов

Филиал СПбГЭУ в г. Чебоксары, Россия

В монографии [1] обсуждается язык объектно-ориентированного имитационного моделирования UML SP (UML Scientific Profile) и методология разработки объектных моделей Modeling SP, использующая этот язык. Профиль UML SP построен на базе метамодели языка UML 1.5. В настоящее время ведется разработка вариации UML SP на базе метамодели UML 2. Кардинальное отличие этого варианта – существенно более структурированная метамодель языка. Для проверки корректности языка потребовалось подготовить банк тестовых моделей и тестовых задач. В данном докладе представлен краткий обзор этой подборки.

Банк моделей состоит из двух групп. В первую группу входят типовые модели имитационного моделирования, среди которых мы укажем следующие: «Микроволновая печь», «Железнодорожная ветка с односторонним движением», «Производственная линия», «Нерегулируемый пешеходный переход». Вторую группу составляют модели научного назначения, среди которых можно назвать: «Коммуникативный акт с рефлекслирующими агентами», «Игры поиска», «Простейший

многоклеточный организм», «Модель Изинга». Имитационные модели реализованы в Borland C++ Builder. Для того, чтобы дать представление об этих моделях с содержательной стороны, рассмотрим одну из них - модель «Микроволновая печь».

Модель микроволновой печи подробно разобрана в книге [2]. Мы включили данную модель в банк моделей потому, что эта модель рассмотрена в книге [3], что позволяет сравнить наш подход с традиционным объектно-ориентированным моделированием. Кроме того, эта модель детально рассмотрена в литературе по темпоральной логике CTL*. Имитационная модель включает «Research Use Case Model», «Research Analysis Model» и «Research Design Model». Модель прецедентов соответствует спецификации, компактно изложенной в [3]. Модель анализа содержит четыре диаграммы классов и две диаграммы взаимодействия. Проектная модель содержит две диаграммы деятельности, одна из которых определяет алгоритм поведения микроволновой печи.

Остановимся подробнее на «Research Analysis Model» (см. рис.1). В отличие от начальных фаз построения модели, мы придерживаемся формулы «Моделирование как программирование», т.е. имена классов и переменных должны отражать вычислительную семантику. С точки же зрения предметной семантики эта диаграмма, как и другие диаграммы UML SP, представляет собой семантическую сеть, концепты которой, определены фреймами.

Модель микроволновой печи является в некотором смысле базовой, т.к. в ней рассмотрен ряд типовых решений. В частности, модель демонстрирует два типовых решения - композицию паттернов *Composite* и *Bridge* и взаимодействие экспериментальной установки с объектом исследования, - которые требуют углубленного анализа.

Мы допустили, что поведение микроволновой печи зависит от того, в каком месте установлена печь (например, есть такие места, где подключение к электросети невозможно). Это позволило на простом примере продемонстрировать довольно сложный вопрос об определении конфигурационного пространства системы (т.е. контейнера для подсистем и атомарных объектов). Согласно методологии Modeling SP объектно-темпоральная декомпозиция имитационной модели описывается паттерном *Composite* [1]. Этот паттерн на рис.1 образуют классы *Component* (в роли *Component*), *Composite* (в роли *Composite*), его потомки (*Cooking*, *Kitchen*) и *Stove* (в роли *Leaf*). Паттерн позволяет единообразно вызывать операцию *Exist* как для узлов дерева, так и для листьев.

происходит вырождение конфигурационного пространства. В предельном случае класс Component становится контейнером для самого себя. Для моделирования более сложных пространств предложено использовать паттерн *Bridge* (см. [1]). Для того, чтобы образовать композицию паттернов *Composite* и *Bridge*, предлагается следующее. Материализуем отношение агрегации между классами Component и Composite посредством класса Knot, тогда класс Composite можно идентифицировать как *Abstraction*, класс Knot как *Implementor*, а отношение композиции определит зависимость «*Categorization*» пакета «*Taxonomy*» от пакета «*Meronomy*». Данное решение позволяет строить довольно сложные неоднородные пространства. Для этого достаточно создать подходящую иерархию наследования для потомков класса Knot. Именно это решение показано на рис.1. Этот пример демонстрирует и другие нетривиальные аспекты. Один из них – взаимосвязь дискретно-событийного времени с конфигурационным пространством. Метод ExistImpr классов-потомков Knot просто вызывает метод Exist потомков класса Component в данной модели. Предложенное решение позволяет создавать более гибкие конструкции; например, в классах-потомках Place можно задать иной темп дискретно-событийного времени, что существенно для моделирования больших и протяженных систем (эффекты неравномерного развития или десинхронизации бизнес-процессов).

Модель экспериментальной установки и обмен сообщениями между экспериментальной установкой и объектом исследования в рассматриваемой имитационной модели представлены в наиболее простом варианте, который, однако, часто оказывается достаточным. Экспериментальная установка состоит из интерфейса пользователя (VCL-класс TMainForm), сенсора EDataProcessor и эктора EActionProcessor. Заметим, что поле _stove класса Cooking так же является частью экспериментальной установки (а не системы). Экспериментатор выбирает нужное действие, экземпляр EActionProcessor считывает из TLabel код действия, декодирует, и совершает над объектом класса Stove соответствующую манипуляцию; экземпляр EDataProcessor считывает состояние печи, кодирует и пересылает эти данные на сцену, которые затем добавляются в TMemo. Чтобы обеспечить пересылку данных, мы допускаем контролируемое нарушение инкапсуляции класса TMainForm. В C++ это достигается директивой #include "Unit1.h" (этот модуль содержит описание TMainForm). В ряде случаев приходится использовать более сложные объектные конструкции, основанные на паттернах Observer и Strategy. Эти решения демонстрируются на похожей, но

другой модели, известной в литературе как «Дом с привидениями» (по У.Р.Эшби).

В научных моделях основное внимание уделяется методологическим вопросам. Например, в «Модель Изинга» проводится четкая грань между численными и имитационными моделями в физике. В имитационных моделях нигде не должны использоваться числовые переменные; все числовые значения должны получаться только как результат измерений. Показательно, что в научных моделях измерительные процедуры и модели экспериментальных установок обычно сложнее самой изучаемой системы.

Предполагается опубликовать описанные выше модели в виде структурированной базы знаний. В состав этой базы будет входить: профиль UML SP в формате XML, краткое описание метамодели профиля, руководство пользователя по UML SP и перечисленные выше модели, которые теперь рассматриваются как демонстрационные. Каждая модель будет представлена: самой моделью в формате XML и XMI, текстовым описанием и документированным кодом на языке C++.

Литература

1. Гурьянов В. И. Имитационное моделирование на UML SP: монография / В.И. Гурьянов.– Чебоксары : Филиал СПбГЭУ в г. Чебоксары, 2014. – 135 с. (см. <http://simulation.su/static/ru-books.html>)
2. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях: Пер. с англ. - Киев: Диалектика, 1993. – 240 с.
3. Труб И.И. Объектно-ориентированное моделирование на C++: Учебный курс – СПб.: Питер, 2006. – 411 с.

УДК 519.237.5

РЕГРЕССИОННЫЙ АНАЛИЗ. ПРОЦЕССНЫЙ ПОДХОД

С.Н. Лапач

Национальный технический университет Украины «КПИ», Киев, Украина

Многие проблемы в использовании планирования эксперимента и регрессионного анализа (РА) связаны с тем, что они представляют собой набор отдельных теорий и методов, не имеющих единства [1,2]. Попытки представить их в виде некоторой технологии построения эмпирических моделей существуют, например [3, 4]. Но подтверждением ее служит только длительное практическое использование. Предлагается с целью разрешения указанного противоречия рассматривать построение регрессионных моделей как часть выборочного метода [2]. Тогда требования к матрице (выборке) и метода обработки получаются