
UML: ИСТОРИЯ, СПЕЦИФИКАЦИЯ, БИБЛИОГРАФИЯ

Дмитрий Буй, Елена Шишацкая

Резюме: *Сделан короткий экскурс в историю возникновения и развития современного универсального языка моделирования UML, приведен короткий обзор языка и предпринята попытка систематизировать научно-методическую литературу, посвященную этому языку.*

Ключевые слова: *UML, объектно-ориентированное проектирование, спецификация, семантика.*

ACM Classification Keywords: *E.4 Coding and information theory – Formal models of communication*

Введение

В настоящее время все большую популярность приобретает язык моделирования UML (Unified Modeling Language), который стал “стандартом де-факто” в области разработки программного обеспечения и применяется для решения задач других предметных областей, например, задач бизнес-моделирования. В работе сделан короткий экскурс в историю возникновения и развития UML, приведен короткий обзор языковых структур и предпринята попытка систематизировать соответствующую библиографию.

Возникновение и назначение языка

В основе UML лежит несколько объектно-ориентированных методов, каждый из которых первоначально был ориентирован на поддержку отдельных этапов *объектно-ориентированного анализа и проектирования* (ООАП) [1]:

- метод Гради Буча (Grady Booch), условное название Booch (Booch'91, Booch Lite, Booch'93) – считался наиболее эффективным на этапах проектирования и разработки программных систем [1];
- метод Джеймса Румбаха (James Rumbaugh), Object Modeling Technique (OMT, позже OMT-2) – оптимально подходил для анализа процессов обработки данных в информационных системах [2];
- метод Айвара Джекобсона (Ivar Jacobson), Object-Oriented Software Engineering (OOSE) – содержал средства представления вариантов использования, имеющих существенное значение на этапе анализа требований в процессе проектирования бизнес-приложений [3].

История развития UML датируется 1994 г., когда началась интеграция/унификация вышеуказанных методов. Проект унифицированного метода (Unified Method) версии 0.8 был опубликован в октябре 1995 г. Все вопросы разработки и сопровождения языка UML сконцентрированы в рамках консорциума OMG (Object Management Group) [4]. Хотя OMG был создан с целью разработки предложений по стандартизации объектных и компонентных технологий CORBA, язык UML приобрел статус второго стратегического направления в работе консорциума. В ноябре 1997 г. OMG объявил UML стандартным языком объектно-ориентированного моделирования и принял на себя обязанности по его последующему развитию. Группа специалистов обеспечивает публикацию описаний последующих версий языка UML и запросов предложений RFP (Request For Proposals) по его стандартизации. Статус языка UML определен как открытый для всех предложений по доработке и усовершенствованию. В 2003 г. в качестве результата рассмотрения набора запросов RFP 2000 г. было опубликовано описание языка UML 2.0, включающее

инфраструктуру UML [5], язык ограничений объектов (Object Constraint Language – OCL) [6], суперструктуру UML [7] и формат обмена диаграмм [8]. Основными инициативами консорциума OMG в рамках работы над проектом UML являются: (1) моделирование систем реального времени; (2) определение модели выполнения – точной спецификации поведения моделей, поддерживающихся UML; (3) обработка данных предприятия – определены так называемые профили, описывающие способы создания больших распределенных параллельных систем предприятия; (4) определение процесса разработки программного обеспечения – специфицированы структуры определения процессов разработки программного обеспечения; (5) стандарт хранения данных; (6) сопоставление технологии CORBA и языка UML; (7) формат XMI (Metadata Interchange) для обмена моделями UML в текстовом формате [9].

Существует консорциум партнеров UML (Digital Equipment Corp., HP, Intellicorp, IBM, ICON Computing, Microsoft, Oracle, Rational Software и другие), обеспечивающих уточнение нотации, усовершенствование и дополнение языка, а также сопровождение разработки инструментальных средств поддержки. Особое место занимает компания Rational Software Corporation [10], реализовавшая Rational Rose 98 – одно из первых инструментальных CASE-средств, в котором был поддержан язык UML [11, 12]. Необходимо отметить внимание компании Microsoft к технологии, базирующейся на языке UML, на основе которой создана информационная модель (UML Information Model), предназначенная для создания стандартного интерфейса между средствами разработки приложений и средствами визуального моделирования. На данный момент Rational Software Corporation официально входит в состав IBM (<http://www-306.ibm.com/software/rational/>).

Язык UML предназначен для решения следующих заданий [13]: (1) предоставить в распоряжение пользователей готовый к использованию выразительный мощный язык визуального моделирования, позволяющий разрабатывать осмысленные модели и обмениваться ими; (2) предусмотреть внутренние механизмы расширяемости и специализации базовых концепций языка; (3) обеспечить максимальную независимость проекта создания программного обеспечения от конкретных языков программирования и процессов разработки; (5) обеспечить формальную основу для однозначной интерпретации языка; (6) стимулировать расширение рынка объектно-ориентированных инструментальных средств создания программного обеспечения; (7) интегрировать лучший практический опыт использования языка и реализации программных средств его поддержки.

Принципы использования UML специфицированы в Rational Unified Process (RUP) – развитой методике создания программного обеспечения, оформленной в виде базы знаний, физически размещенной на web (<http://www-306.ibm.com/software/awdtools/rup/support/>) и оснащенной поисковой системой [14, 15]. Этот коммерческий продукт задает строгий регламент распределения задач и ответственности между исполнителями в процессе разработки программного обеспечения.

Основным документом по UML является [16], где описана *метамодель* UML и очень мало внимания уделяется семантике языковых конструкций. Описание текущей версии языка UML [5-8] и примеры разработки программных систем с использованием CASE-средства Rational Rose можно найти на web-узле OMG (www.omg.org); модификации языка UML и его новейшие версии – на узле www.celigent.com/uml.

Так как официальная документация по UML весьма затруднительна для понимания, то выходит много книг, описывающих язык с разными акцентами. Например, базовая система обозначений UML популярно и доступно изложена в книге Мартина Фовлера (Martin Fowler) [17], которая считается одним из наилучших пособий по изучению языка. Следует отметить книги, написанные главными авторами UML – Г. Бучем, И. Джекобсоном, Д. Рембо (в другом переводе Румбахом или Рамбо): в [13] детальная информация об

использовании UML (покрывает около 80% языка) проиллюстрирована применением языка на большом количестве примеров; в [15] рассматривается процесс объектно-ориентированной разработки программного обеспечения; в справочнике по UML [18] охватывается весь язык и делается попытка раскрыть его содержательную семантику. Начинаящим стоит обратить внимание на издание [19], вышедшее в серии "... для чайников"; детальное и полное описание языка также приведено в [20]. Вопросам использования продукта Rational Rose посвящены [21, 22], но наиболее развернуто и доступно моделирование систем средствами UML с использованием Rational Rose описано и проиллюстрировано большим количеством примеров в [23] (переиздание [24] – расширено и адаптировано под UML 2.0). Также необходимо выделить серию книг под редакцией главных авторов языка: "Серия объектные технологии" издательства Addison Wesley посвящена применению языка [14, 25, 26, 27, 28]; применению UML и шаблонов проектирования посвящены также [29, 30].

UML создавался как язык моделирования общего назначения для применения в таких "дискретных" областях, как программное обеспечение, аппаратные средства и цифровая логика. Структуры UML позволяют фиксировать разнообразные решения по отображению (1) функциональности системы [28, 31, 32], (2) динамической и статической структуры системы, (3) организации элементов системы [33], (4) реализации системы. Популярность приобретает использование UML при проектировании баз данных [26, 34, 35]. Благодаря открытости (наличию в языке механизмов расширения) он предоставляет мощный инструментарий для решения задач других отраслей, например, бизнес-моделирования [36].

Общая структура языка

Семантика языка UML определяется для двух видов объектных моделей: *структурных* и *поведенческих*. Структурные (статические) модели описывают структуру сущностей или компонентов системы, включая их *классы, интерфейсы, атрибуты* и *связи*. Модели поведения (динамические) описывают поведение или функционирование объектов системы, включая их *методы, взаимодействие (сотрудничество)* между ними, а также процесс изменения состояний отдельных компонентов и системы в целом [1].

Формальное описание языка UML основывается на следующей общей иерархической структуре модельных представлений, состоящей из четырех уровней абстракции: (1) мета-метамодель, (2) метамодель, (3) модель, (4) объекты пользователя [5]. Уровень мета-метамодели образует базовую основу для всех метамодельных представлений и определяет язык для спецификации метамодели. Мета-метамодель определяет модель языка UML на наивысшем уровне абстракции (соответственно на самом низком уровне конкретизации) и является наиболее компактным его описанием. Метамодель – экземпляр или конкретизация мета-метамодели – определяет язык для спецификации моделей. Все основные понятия языка UML – это понятие уровня метамодели. Модель в контексте языка UML является экземпляром (конкретизацией) метамодели в том понимании, что любая (конкретная) модель системы должна использовать только понятия метамодели, конкретизировав их применительно соответствующей ситуации. Содержательно говоря, уровень модели предназначен для описания конкретной предметной области. Конкретизация понятий модели происходит на уровне объектов, которые являются экземплярами модели и содержат конкретную информацию о предметной области в соответствии с понятиями модели.

Основой представления UML на метамодельном уровне является описание трех его *логических блоков (пакетов)*: *основные элементы, элементы поведения* и *общие механизмы* [7].

Концептуальная модель языка

Концептуальная модель языка включает основные строительные блоки, правила их сочетания и общие механизмы [13, 17, 18].

Словарь языка UML содержит *сущности* (абстракции, являющиеся основными элементами модели) и *отношения* (основные соединительные строительные блоки), Сущности и отношения по определенным правилам соединяются в конструкции – *диаграммы*.

В UML определено четыре типа сущностей [13]:

– *структурные сущности*, распадающиеся на *основные* (*класс* (Class), *интерфейс* (Interface), *кооперация* (Collaboration), *прецедент* (Use case), *активный класс* (Active class), *компонент* (Component), *узел* (Node)), *разновидности основных* (*актер* (Actor), *сигнал* (Signal), *утилита* (Utility, вид классов), *процесс* (Process), *нить* (Thread, вид активных классов)) и *остальные* (*приложения* (Application), *документ* (Document), *файл* (File), *библиотека* (Library), *страница* (Page), *таблица* (Table, вид компонентов));

- *сущности поведения* (Behavioral things) – *взаимодействие* (Interaction) и *автомат* (State machine);
- *группирующие сущности* – *пакет* (Packages);
- *анотационные сущности* – *примечание* (Note).

Основными типами отношений в UML являются отношения: *зависимости* (Dependency), *ассоциации* (Association) (разновидностью ассоциации является отношение *агрегации* (Aggregation)), *обобщения* (Generalization) и *реализации* (Realization). Существуют также их вариации, например, *уточнение*, *трассировка*, *включение* и *расширение* (для отношений зависимости).

Для построения корректно оформленной модели в UML определены правила, позволяющие корректно и однозначно определять: (1) имена сущностей, отношений и диаграмм, (2) область действия имен (контекст, в котором имя имеет некоторое значение), (3) видимость имен (для использования другими элементами), (4) целостность (правильность и согласованность соотношения элементов), (5) выполнение модели [13].

Эффективность и упрощение применения языка обеспечивается использованием определенных соглашений, так называемых, *общих механизмов*: *спецификаций* (Specifications), *дополнений* (Adornments), *принятых распределений* (Common divisions) и *механизмов расширения* (Extensibility mechanisms) [13, 17, 18].

Каждый элемент нотации UML имеет уникальное графическое обозначение и спецификацию – текстовое представление синтаксиса и содержательной семантики соответствующего строительного блока. Практически все строительные блоки характеризуются дихотомией “класс / объект” и “интерфейс / реализация”. Это основные подходы деления реальности при объектно-ориентированном моделировании систем.

UML допускает контролируемые расширения для адаптации языка к конкретным потребностям. Наличие внутренних механизмов расширения принципиально отличает UML от таких средств моделирования как IDEF0, IDEF1X, IDEF3, DFD и ERM [37], являющихся замкнутыми и не допускающими расширения средствами самого языка. К *механизмам расширения* UML относятся: (1) *стереотип* (Stereotype), расширяющий словарь языка (позволяет создавать из существующих блоков новые, специфические для конкретной решаемой задачи); (2) *тэг-значение* (Tagged value), расширяющее свойства строительного блока (дает возможность включать новую информацию в *спецификацию* элемента); (3) *ограничение* (Constraint), расширяющее семантику строительного блока (позволяет добавлять новые или

модифицировать существующие правила посредством семантических ограничений, заданных на естественном языке либо на формальном языке OCL). Некоторые расширения приобрели такую популярность, что были внесены в стандарт текущей версии UML [13, 17, 18].

Графические конструкции языка

В нотации UML все представления о моделях сложной системы фиксируются в виде специальных графических конструкций, получивших название *диаграмм* [6]. Диаграмма в UML – это графическое представление набора элементов, изображаемое, как правило, в виде связанного графа с вершинами (сущностями) и ребрами (отношениями). Теоретически диаграммы могут содержать любые комбинации сущностей и отношений. Однако на практике применяется небольшое количество типовых комбинаций. В языке UML определены следующие виды диаграмм: (1) *диаграмма вариантов использования* (Use case diagram); (2) *диаграмма классов* (Class diagram); (3) *диаграммы поведения* (Behavior diagrams), к которым относятся *диаграмма состояний* (Statechart diagram), *диаграмма деятельности* (Activity diagram), *диаграммы взаимодействия* (Interaction diagrams); в свою очередь к диаграммам взаимодействия относятся *диаграмма последовательности* (Sequence diagram) и *диаграмма кооперации* (Collaboration diagram); (4) *диаграммы реализации* (Implementation diagrams), к которым относятся *диаграмма компонентов* (Component diagram) и *диаграмма развертывания* (Deployment diagram).

Перечень этих диаграмм и их названий является каноническим, они – неотъемлемая часть графической нотации языка UML. Процесс ООАП неразрывно связан с процессом построения этих диаграмм, при этом совокупность построенных диаграмм является самодостаточной в том понимании, что в них содержится вся информация, необходимая для реализации проекта сложной системы.

Каждая из диаграмм детализирует и конкретизирует разные представления о моделях сложной системы в терминах языка UML. При этом диаграмма вариантов использования является наиболее общей концептуальной моделью сложной системы, исходной для построения всех других диаграмм. Диаграмма классов есть, по сути, *логической моделью*, отображающей *статические аспекты структурного построения* сложной системы. Диаграммы поведения также являются разновидностями логической модели, которые отображают *динамические аспекты функционирования* сложной системы. И, наконец, диаграммы реализации служат для представления физических компонентов сложной системы и поэтому относятся к ее *физической модели*. Таким образом, интегрированная модель сложной системы в нотации UML представляется в виде совокупности отмеченных выше диаграмм.

Следует обратить внимание, что при моделировании системы часто строят *диаграмму объектов* (Object diagram), представляющую множество объектов и отношений между ними в заданный момент времени (“снимок системы”); не являясь канонической диаграммой UML, эта диаграмма полезна в детализации системы.

Подходы к спецификации семантики языка

Начальной целью развития UML было обеспечение более точного описания языка моделирования – поддержка формальной основы для понимания языка моделирования. Однако до настоящего времени формальная семантика не является частью стандарта. Обзор нескольких подходов, касающихся определения такой семантики, приведен в [38], где рассмотрены *теоретико-множественный, трансляционный, метамодельный* подходы и предложена так называемая “*свободная семантика*”. Следует отметить, что в самой спецификации UML существует много огрехов и противоречий: так, в работе [39] рассматриваются отношения включения и расширения для вариантов использования, в [40]

анализируются стандартные стереотипы метамодели UML. В работе [41] предлагается использовать RM-ODP (Reference Model Open Distributed Processing [42]) для решения проблем языка. Отметим, что согласно спецификации UML модель RM-ODP выступает структурой, которая непосредственно влияет на архитектуру метамодели языка (раздел “Preface: Relationships to Other Models” [7]). Кроме того RM-ODP используется в MOF (Meta-Object Facility) для управления типами. В [41] идентифицируются три проблемы метамодели UML и предлагается их решение на базе RM-ODP:

– структурный хаос семантики языка – “высокая техника, лаконичность и сложность для понимания новичками”; решение: использование структуры RM-ODP и теории типов Б. Расела;

– отсутствие декларативной семантики, противоречивость семантики языка при описании (1) отношений между моделями, построенными с использованием языка, и (2) непосредственно субъектов моделирования; решение: реализация базовой концепции моделирования (Basic Modeling Concept);

– недостаточное теоретическое обоснование используемой метамодели языка UML; решение: предлагаемая в статье концепция моделирования на основе RM-ODP, теории типов Б. Расела и подхода А. Тарского к интерпретации формальных исчислений считается полностью обоснованной.

В многих работах, посвященных формализации модели и метамодели языка UML, рассматривается не сам язык, а некоторый его подязык, формальный и строго структурированный. Так, в [43] рассматриваются BON (Business Object Notation, объектно-ориентированный язык моделирования, по существу совпадающий с подязыком диаграмм классов UML [44]) и PVS (Prototype Verification System, язык спецификаций, разработанный для автоматического анализа метамоделей языков моделирования [45]). Результатом этой работы является полная формальная спецификация метамодели объектно-ориентированного языка моделирования в форме, пригодной для автоматического анализа. Однако BON по сравнению с UML более формализованный и “подогнанный” под условия решаемой задачи. Аналогичный подход использован в [46], где в качестве платформы для формализации выбран формализм Boon, состоящий из метамодели и языка формальных спецификаций Odal – простого строго типизированного языка, семантика которого задана в терминах так называемого π -исчисления. В [47] на основе [48] рассматривается формализация языка MML (Metamodelling Language), являющегося подмножеством UML; эта формализация предложена авторами в качестве базы для всего UML 2.0. Наконец, в работе [49] демонстрируется применимость алгебраического подхода для формального описания ER-диаграмм (Entity-Relationship diagrams), являющихся аналогами диаграмм классов UML.

Выводы

UML является мощным, гибким средством моделирования, описание стандарта которого является открытым для последующего совершенствования. Неоднозначность как некоторых конструкций самого языка, так и подходов к его формальной семантике, наличие в спецификации неформальных описаний требует дальнейшего развития формальной основы для полной и непротиворечивой интерпретации языка.

Литература

Booch G. Object-oriented analysis and design with applications. Second edition. – The Benjamin/Cummings Publishing Company Inc., 1994. – 589 p.

Rumbaugh J., Blacha M., Premerlani W., Eddy F., Lorenzen W. Object-Oriented Modeling and Design. – Prentice-Hall Inc., 1991. – 432 p.

- Jacobson I. Object-Oriented Software Engineering. A Use Case Driven Approach. – Addison-Wesley Publishing Company, 1993. – 547 p.
- Object Management Group, 2003. OMG Unified Modeling Language Specification / www.omg.org.
- www.omg.org / 04-10-14pdf.
- www.omg.org / 05-06-06pdf.
- www.omg.org / 05-07-04pdf.
- www.omg.org / 05-06-04pdf.
- Рамбо Д. Тенденции в развитии языка UML и разработки ПО / <http://www.interface.ru/fset.asp?Url=/rational/umltend.htm>.
- <http://www.rational.com/uml>.
- Кумсков М. Rational Rose 98 – CASE-продукт нового поколения / <http://www.interface.ru/public/rose98/rose98.htm>.
- Новичков А. Rational Rose для разработчиков / <http://www.interface.ru/fset.asp?Url=/rational/xros.htm>.
- Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. – MA.: Addison-Wesley Publishing Co., 1999. – 512 p.
- Kruchten P. The Rational Unified Process – An Introduction. – MA.: Addison-Wesley, 2000. – 234 p.
- Jacobson I., Booch G., Rumbaugh J. The Unified Software Development Process. – MA.: Addison-Wesley Publishing Co., 1999. – 512 p.
- Booch G., Jacobson I, Rumbaugh J. The UML specification documents. – Santa Clara, CA.: Rational Software Corp., 1997. (Спецификация размещена на www.rational.com.)
- Fowler M., Scott K. UML Distilled. – MA.: Addison-Wesley, 2000. – 472 p.
- Rumbaugh J., Jacobson I., Booch G. Unified Modeling Language Reference Manual. – MA.: Addison-Wesley Publishing Co., 1999. – 576 p.
- Chonoles M. J., Schardt J.A. UML 2 for Dummies. – Hungry Minds, 2003. – 412 p.
- Kendal S. Fast Track UML 2.0. – Apress, 2004., 416 p.
- Трофимов С. А. Практическая работа в Ration Rose / <http://oad.asf.ru/Files.aspx>
- Вендров А. М.. Объектно-ориентированный анализ и проектирование с использованием языка UML и Rational Rose. Практикум / <http://oad.asf.ru/Files.aspx>
- Boggs W., Boggs M. Mastering UML witch Rational Rose. – Sybex Inc., 1999. – 579 p.
- Boggs W., Boggs M. UML witch Rational Rose 2.0. – Sybex Inc., 2002. – 702 p.
- Кватрани Т. Визуальное моделирование с помощью Rational Rose и UML: Пер. с англ. – Москва: Издательский дом “Вильямс”, 2003. – 192 с.
- Нейбург Э., Максимчук Р. Проектирование баз данных с помощью UML.: Пер. с англ. – Москва: Издательский дом “Вильямс”, 2002. – 288 с.
- Гома Х. UML. Проектирование систем реального времени, распределенных и параллельных приложений.: Пер. с англ. – Москва: ДМК, 2002. – 704 с.
- Розенберг Д., Скотт К. Применение объектно-ориентированного моделирования с использованием UML и анализ прецедентов.: Пер. с англ. – Москва: ДМК, 2002. – 160 с.
- Ларман К. Применение UML и шаблонов проектирования. 2-е издание.: Пер.с англ. – Москва: Издательский дом “Вильямс”. 2004. – 604 с.
- Gamma E., Helm R., Jonson R., Vissides J. Element of Reusable Object-Orient Software. – Addison Wesley Longman Inc., 1994. – 372 p.

- Коберн А. Современные методы описания функциональных требований к системам.: Пер. с англ. – Москва: ЛОРИ, 2002. – 262 с.
- Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход.: Пер. с англ. – Москва: “Вильямс”, 2002. – 448 с.
- Duffy Daniel J. Domain architectures: models and architectures for UML applications. – John Wiley & Sons Ltd, 2004. – 390 p.
- Мюллер Р. Базы данных и UML. Проектирование.: Пер. с англ. – Москва: ЛОРИ, 2002. – 420 с.
- Nock C. Data Access Patterns: Database Interactions in Object-Oriented Applications. – Addison Wesley, 2003. – 512 p.
- Eriksson H.-Erik, Penker M. Business Modeling with UML: Business Patterns at Work. – Wiley Computer Publishing, 2000. – 274 p.
- Маклаков С. В. Создание информационных систем с AllFusion Modeling Suite. – Москва: ДИАЛОГ-МИФИ, 2005. – 432 с.
- Husman H. Loose Semantics for UML/OCL // Society for Design and Process Science, 2002. – P. 32-39.
- Genova G., Llorens J., Quintana V. Digging into Use Case Relationships // Lect. Notes Comput. Sci. – 2002. – V. 2460. – P. 115–127.
- Gogolla M., Henderson-Sellera B. Analysis of UML Stereotypes within the UML Metamodel // Lect. Notes Comput. Sci. – 2002. – V. 2460. – P. 84-99.
- Naumenko A., Wegmann A. A Metamodel for the Unified Modeling Language // Lect. Notes Comput. Sci. – 2002. – V. 2460. – P. 2-17.
- RM-ODP Open Distributed Processing – Reference Model - ISO, ITU.: ISO/IEC 10746-1, 2, 3, 4 | ITU-T Recommendation X.901, X.902, X.903, X.904, 1995-1998.
- Paige R., Ostroff J. Metamodelling and Conformance Checking with PVS // Lect. Notes Comput. Sci. – 2001. – V. 2029. – P. 2-16.
- Walden K., Nerson J.-M. Seamless Object-Oriented Software Development. – Prentice-Hall, 1995.
- Owre S., Shankar N., Rushby J., Stringer-Calvert D. The PVS Language Reference Version 2.3 – September, 1999. – (Tech. Rep. / SRI International Technical Report).
- Overgaard G. Formal Specification of OO Modeling // Lect. Notes Comput. Sci. – 2000. – V. 1783. – P. 193-207.
- Clark T., Evans A., Kent S. The Metamodelling Language Calculus: Foundation Semantics for UML // Lect. Notes Comput. Sci. – 2001. – V. 2029. – P. 17-31.
- Cardeli L, Abadi M. A theory of Objects. – Springer-Verlag, 1996.
- Lellahi K. Conceptual Data Modeling: An Algebraic Viewpoint // Lect. Notes Comput. Sci. – 2001. – V. 2244. – P. 336-348.

Информация об авторах

Буй Дмитрий Борисович – заведующий лабораторией проблем программирования, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики: Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: buy@unicyb.kiev.ua

Шишацкая Елена Владимировна – инженер-программист лаборатории проблем программирования, Киевский национальный университет имени Тараса Шевченко, факультет кибернетики: Украина, Киев, 03680, пр. Глушкова 2, корп.6; e-mail: shyshatskaja@unicyb.kiev.ua