

УДК 519.7: 007 + 06

О.В. ОЛЬХОВИК

ПРОЕКТНЫЕ ДИАГРАММЫ НА N-VISUAL LANGUAGE И МОДЕЛИРОВАНИЕ БИЗНЕС-ПРОЦЕССОВ

Показано, какие проблемы позволяют решать диаграммы на N-Visual Language, предназначенные для проектирования и автоматической кодогенерации программного обеспечения информационных систем. Рассмотрен вопрос связывания проектных диаграмм на N-Visual Language с диаграммами, моделирующими бизнес-процессы, в различных нотациях.

Ключевые слова: моделирование бизнес-процессов, информационная система, CASE-технологии, Model-driven engineering, визуально-декларативный язык, проектирование программного обеспечения.

Введение. В области производства программного обеспечения информационных систем (ПО ИС) существует ряд проблем, которые приводят, по разным оценкам, к неудаче от 25 до 35% всех проектов. При этом финансовые потери во всем мире составляют миллиарды долларов. Кроме того, существуют проблемы сопровождения ПО ИС в средних и крупных организациях, приводящие к постоянному повышению сложности этого процесса. Организации вынуждены последовательно увеличивать расходы на поддержку внедренных информационных систем. Такая ситуация приводит к тому, что либо расходы на сопровождение информационных систем начинают превышать получаемый от них экономический эффект, либо приходится отказываться от таких расходов, в результате чего внедренные информационные системы нередко коллапсируют.

Механизм связывания проектных и аналитических диаграмм. Проблемы ПО ИС во многом обусловлены несовершенством используемых при производстве программного обеспечения технологий. Современные Model-driven engineering технологии (MDE-технологии) предназначены для улучшения коммуникаций между участниками проекта и для генерации программного кода на основе проектных диаграмм. Тяжелые методологии разработки ПО ИС, такие, как Rational Unified Process или Structured Analysis and Design Technique, предполагают широкое использование MDE-технологий. Однако они повышают стоимость разработки и сопровождения программ из-за необходимости затрат отдельно на разработку (поддержку) проектных диаграмм и отдельно на создание (изменение) программного кода. Часто это приводит к несоблюдению методологий на практике и к расхождению проектных диаграмм и программного кода в процессе эксплуатации.

По мнению многих исследователей, MDE-технологии неэффективны для малых проектов, а для средних и больших проектов их эффективность намного ниже ожидаемой. Большинство новых «гибких» методологий разработки программ (Extreme Programming, Crystal, Adaptive Software Development) либо вообще отрицают необходимость MDE-технологий, либо допускают их использование только в коммуникативном аспекте. Причина такого положения – невозможность полной автоматической генерации программного кода из проектных диаграмм. Существующие нотации, например Executable UML, базирующиеся на стандарте Unified Modeling Language 2.0 (UML 2.0), позволяют генерировать код лишь частично. При этом нотации настолько усложняются, что становятся понятными не более чем просто хорошо откомментированная программа. Тем самым их коммуникативная ценность сводится к нулю – проектные диаграммы непонятны и бесполезны специалистам в предметных областях, а их разработка не менее проста, чем построение кода.

Стоимость сопровождения ПО ИС также может быть существенно снижена за счет увеличения числа запросов «ad hoc», выполняемых пользователями без помощи программистов. Сейчас эта задача решается технологией интерактивных отчетов (Interactive Reports), однако ее возможности ограничены тем, что расширить исходный набор атрибутов в отчете можно только посредством перестраивания запроса. Последнее, как правило, требует привлечения программистов. Увеличить число запросов «ad hoc», выполняемых пользователями, можно, если упростить

Structured Query Language (SQL) или, точнее, заменить его на более простой SQL-подобный язык. Чтобы не терять выразительной мощности, такой язык должен основываться на нереляционной модели данных. Наиболее популярными из постреляционных являются модели, соответствующие стандарту Object Data Management Group 3.0. Однако этот стандарт не ставит целью упрощение языка запросов, и поэтому работы по созданию декларативных ненавигационных языков ведутся вяло.

Можно выстроить прямую цепочку: проблемы разработки и сопровождения ПО ИС – недостатки тяжелых методологий – недостатки визуальных нотаций (UML 2.0 или Integrated Definition) – недостатки моделей данных (реляционной и объектных). Эта цепочка демонстрирует, что для решения практических проблем необходимо последовательное решение задач, начиная с разработки новой модели данных.

В работе [1] представлена N-модель данных, на основе которой разработаны декларативный язык запросов N-Declarative Language [2] и визуально-декларативный язык проектирования N-Visual Language (NVL) [3]. Методы создания нормализованных (неизбыточных) проектных диаграмм на языке NVL предложены в работах [4, 5]. Таким образом, разрабатывается принципиально новая MDE-технология, способная конкурировать с существующими:

- за счет сокращения цикла производства и внесения изменений в ПО ИС посредством полной автоматической кодогенерации структур данных и бизнес-логики;
- повышения коммуникативного эффекта благодаря более простому и понятному визуально-декларативному языку моделирования;
- снижения числа обращений в процессе эксплуатации ПО ИС вследствие использования более простого языка запросов, позволяющего большую часть запросов «ad hoc» строить без вмешательства программистов.

Новая технология позволяет снизить стоимость и риски процессов проектирования и разработки ПО ИС. Однако для информационных систем существенны и риски, связанные с анализом предметной области. К таковым, прежде всего, можно отнести следующие:

- разработанная и уже внедренная система решает лишь часть задач, для которых была предназначена;
- разработанная система не соответствует ожиданиям заказчика в том смысле, что решает не те проблемы, которые важны для него в первую очередь;
- разработанная и внедренная система не соответствует принятым у заказчика технологиям обработки данных и документов и поэтому не используется на практике.

Перечисленные риски являются следствием несоответствия реализованного в системе функционала имеющимся у заказчика бизнес-процессам. Это несоответствие может возникать по нескольким причинам. Как правило, у заказчика отсутствует четкое представление о том, какие задачи должна решать информационная система, или же нет согласованного представления об этом на различных уровнях исполнения. Поэтому требования к ПО ИС могут формулироваться расплывчато, а иногда и противоречиво.

Попытка формализовать требования заказчика только посредством технического задания (ТЗ) малоэффективны. Они часто приводят лишь к тому, что у исполнителя появляется возможность оправдаться: что в задании указано, то и сделано. Но почему не указали то, что было нужно? Или не так изложили? Потому что техническое задание – это не тот документ, по которому легко понять, какие потребности заказчика будут реализованы, а какие нет. Структура этого текстового документа не дает целостного представления, позволяющего в дальнейшем углубиться в детали. А самое главное, ТЗ на информационную систему не имеет прямых связей с тем, как и что делается на автоматизируемом предприятии. Иными словами, нет прямой связи между планируемым к реализации функционалом системы и принятыми на предприятии бизнес-процессами. Это и является основной причиной того, что некоторые бизнес-процессы «выпадают» из автоматизации, затрагиваются лишь частично или автоматизируются не так, как представляется заказчику.

Для исключения подобных ситуаций ТЗ должно разрабатываться не на устных положениях заказчика, а на основе результатов анализа бизнес-процессов. При этом результаты такого анализа должны формализовываться в виде, доступном для легкого понимания и заказчиком, и исполнителем. Инструментом такой формализации давно стали CASE-технологии. Как правило, для моделирования бизнес-процессов используются нотация Data Flow Diagram (DFD), функциональные диаграммы IDEF0, диаграммы активности (Activity Diagram) UML 2.0 или диаграммы в нотации Business Process Modeling Notation (BPMN).

Проектные диаграммы на языке NVL для моделирования бизнес-процессов использоваться не могут. В отличие от либерального стандарта UML, где почти каждый элемент языка может участвовать в моделировании всего, что угодно, NVL утверждает строгую стилистику: каждый элемент языка имеет только одно назначение и не может интерпретироваться различными способами.

Тогда каким образом осуществить переход от модели бизнес-процесса к проектным диаграммам? Необходимо связать элементы диаграмм, описывающих бизнес-процессы с элементами проектных диаграмм на языке NVL.

Сначала покажем, как произвести связывание проектных диаграмм с нотацией DFD, затем обобщим его для случаев IDEF0, Activity Diagram и BPMN. N-модель данных, на которой основан язык NVL, дает для этого удобный инструмент. Она постулирует, что атрибут является функцией, определенной на множестве объектов, причем, способы задания этой функции могут быть различными. Атрибут является исходным, если его значения задаются перечислением. Атрибут является расчетным, если его значения определяются последовательностью операций (формулой). Таким образом, в N-модели не делаются принципиальные различия между исходными и расчетными данными, а вычислительные процессы декларативно определяются в расчетных атрибутах.

Обозначим отношение информативности $I \subseteq F \times X$, где F – множество атрибутов, X – множество объектов (в том числе классов). Обозначим символом \mathfrak{I} с индексом множество атрибутов, определенных на классе x с тем же индексом или в его категориях. Согласно [1], атрибут f_i может быть информативен на классе x_i , если:

– определен на этом классе или его категории

$$f_i \in \mathfrak{I}_i \Rightarrow (f_i, x_i) \in I;$$

– определен на классе, от которого он наследует, или его категории

$$f_i \in \mathfrak{I}_j, x_i \subseteq x_j \Rightarrow (f_i, x_i) \in I;$$

– определен на классе или категории класса, который наследует от него

$$f_i \in \mathfrak{I}_j, x_i \supseteq x_j \Rightarrow (f_i, x_i) \in I.$$

Введем отношение композиционной информативности $I' \subseteq F \times X$. Атрибут f_i композиционно информативен на классе x_i , если:

– информативен на этом классе

$$(f_i, x_i) \in I \Rightarrow (f_i, x_i) \in I';$$

– композиционно информативен на классе x_j и класс x_i может быть связан через операцию композиции с классом x_j

$$(f_i, x_j) \in I', \exists (g_i, x_i) \in I, \text{ что } g_i(x_i) \subseteq x_j \Rightarrow (f_i, x_i) \in I'.$$

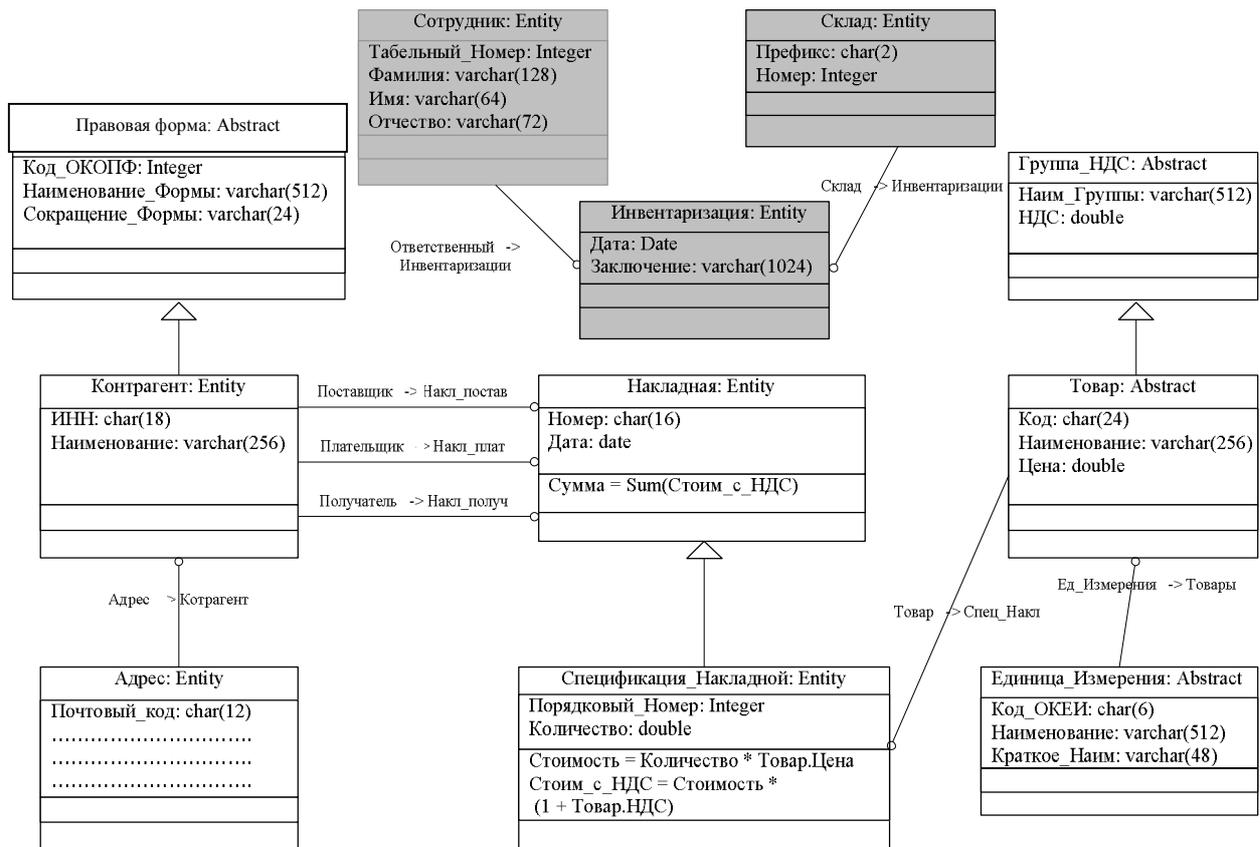
Обозначим множество атрибутов, композиционно информативных на произвольном классе x , символом Ψ с таким же индексом, как у класса:

$$\Psi_i = \{f' / (f, x_j) \in I'\}.$$

Назовем потоком и обозначим символом s подмножество произвольного множества Ψ_i . Множество потоков S определим как

$$S = \{s / \exists x_i, \text{ что } s \in \Psi_i\}.$$

В проектной диаграмме на языке NVL потоку может соответствовать некоторое множество атрибутов связанных между собой классов.



Фрагмент предметной области «Складской учет»

В приведенном на рисунке примере можно выделить следующие множества композиционно информативных атрибутов:

$$\begin{aligned}
 &\Psi_{\text{Накладная}} = \Psi_{\text{Спецификация_Накладной}} = \Psi_{\text{Контрагент}} = \Psi_{\text{Правовая_Форма}} = \\
 &= \Psi_{\text{Товар}} = \Psi_{\text{Группа_НДС}} = \Psi_{\text{Адрес}} = \Psi_{\text{Единица_Измерения}} = \\
 &\{\text{Накладная.Номер, Накладная.Дата, Накладная.Сумма, Специфика-} \\
 &\text{ция_Накладной.Порядковый_Номер, Спецификация_Накладной.Количество,} \\
 &\text{Спецификация_Накладной.Стоимость, Спецификация_Накладной.Стоимость_с_НДС,} \\
 &\text{Контрагент.ИНН, Контрагент.Наименование, Правовая_Форма.Код_ОКОПФ,} \\
 &\text{Правовая_Форма.Наименование_Формы, Правовая_Форма.Сокращение_Формы, Товар.Код,} \\
 &\text{Товар.Наименование, Товар.Цена, Группа_НДС.Наим_Группы, Группа_НДС.НДС,} \\
 &\text{Адрес.Почтовый_Индекс, Единица_Измерения.Код_ОКЕИ, Единица_Измерения.} \\
 &\text{Наименование, Единица_Измерения.Краткое_Наим}\}, \\
 &\Psi_{\text{Сотрудник}} = \Psi_{\text{Склад}} = \Psi_{\text{Инвентаризация}} = \\
 &\{\text{Сотрудник.Номер, Сотрудник.Фамилия, Сотрудник.Имя, Сотрудник.Отчество,} \\
 &\text{Склад.Префикс, Склад.Номер, Инвентаризация.Дата, Инвентаризация.Заключение}\}.
 \end{aligned}$$

Поток, непосредственно формирующий товарную накладную, как документ¹, выделяется как подмножество $\Psi_{\text{Накладная}}$ (или любого равного ему множества композиционно информативных атрибутов):

$$s = \{\text{Накладная.Номер, Накладная.Дата, Накладная.Сумма, Спецификация_Накладной.Порядковый_Номер, Спецификация_Накладной.Количество, Спецификация_Накладной.Стоимость, Спецификация_Накладной.Стоимость_с_НДС, Контрагент.Наименование, Правовая_Форма.Сокращение_Формы, Товар.Код, Товар.Наименование, Товар.Цена, Адрес.Почтовый_Индекс, Единица_Измерения.Краткое_Наим}\}.$$

Определение потока полезно не только для рассматриваемого вопроса, но еще и формально определяет множество атрибутов, значения которых могут быть получены в одном запросе на языке N-Declarative Language. Результатом любого запроса могут быть значения атрибутов только одного потока.

Нотация DFD предоставляет для моделирования бизнес-процессов четыре механизма: процессы, потоки данных, хранилища данных и внешние сущности. Очевидно, что ни процессы, ни внешние сущности нельзя сопоставить с введенным выше понятием потока. Нельзя с ним сопоставить и понятие хранилища данных – последнее более узко, поскольку определяет только сохраняемые исходные данные. Остаются потоки данных, которые определяются как механизмы, используемые для моделирования передачи информации (или даже физических компонентов) из одной части системы в другую.

С одной стороны, понятие потока данных в DFD кажется более широким, чем понятие потока в смысле N-модели, поскольку позволяет моделировать и физические компоненты. Тем не менее необязательно связывать все потоки данных с проектной диаграммой на языке NVL – достаточно ограничиться только теми из них, которые моделируют передачу информации. Общепринятый подход предполагает спецификацию таких потоков с помощью, например, форм Бэкуса – Наура (БНФ). Однако такая спецификация не имеет большой практической ценности – из БНФ нельзя получить ни код, ни структуры данных. Поэтому предлагается специфицировать потоки данных DFD с помощью потоков N-модели. Причем одному потоку DFD соответствует строго один поток N-модели. В контексте проектных диаграмм на языке NVL, потоку данных соответствует множество атрибутов связанных между собой классов.

Предлагаемый подход позволяет осуществить гармоничный переход от диаграмм, моделирующих бизнес-процессы, к диаграммам на языке NVL, из которых генерируются структуры данных и программный код. Иными словами, связывание потоков данных с потоками N-модели выстраивает мостик между анализом и остальными фазами проекта.

В стандарте IDEF0 [6] бизнес-процессы моделируются с помощью функциональных диаграмм. В этой нотации представлены следующие элементы: работы, входы, выходы, управление, механизмы и вызовы. Потокам N-модели здесь соответствуют только входы и выходы. Вход – материал или информация, которые используются или преобразуются работой для получения результата (выхода). Выход – материал или информация, которые производятся работой. Как и в случае с DFD, входы и выходы могут описывать материальные объекты. Но, как и в предыдущем случае, потоки N-модели могут специфицировать только те входы и выходы, которые моделируют информационное взаимодействие.

Использование для моделирования бизнес-процессов Activity Diagram стандарта UML 2.0 [7] хотя и ненормативно, но на практике встречается нередко. Если отбросить такие элементы Activity Diagram, как состояния, активности (действия), дорожки, ворота и объекты, то для связи с потоками N-модели остаются потоки управления (переходы). Специфицируемый поток должен интерпретироваться как информационный и передаваться от одной активности к другой.

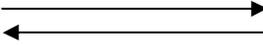
Нотация BPMN [8] для моделирования бизнес-процессов в настоящее время является наиболее перспективной. Она поддерживается концерном OMG и, кроме того, встроена во многие

¹ Из примера в данном случае исключены банковские реквизиты и адреса контрагентов, а также основание и сведения о товарно-транспортной накладной.

коммерческие системы электронного документооборота или управления контентом. Таким образом, описания бизнес-процессов в нотации BPMN используются не только для построения информационных систем, но и для четкого задания правил и ограничений электронного документооборота.

Нотацию BPMN можно рассматривать как некое расширение Activity Diagram. Однако она, в хорошем смысле, сужает «свободу творчества» аналитика, накладывая жесткие ограничения на интерпретацию элементов. Как и в Activity Diagram для моделирования бизнес-процессов, здесь можно использовать состояния, активности (действия), ворота и дорожки. В качестве объектов взаимодействия используются ассоциации, потоки взаимодействия и потоки сообщений. Поскольку последние трактуются как механизмы, показывающие передачу сообщений между участниками бизнес-процессов, для их спецификации можно использовать потоки N-модели (см. таблицу).

Спецификация элементов диаграмм, моделирующих бизнес-процессы
атрибутами проектных диаграмм на языке NVL

Нотация	Элемент	Примечание
Data Flow Diagram (DFD)	Потоки данных 	Специфицируются потоки, моделирующие передачу информации.
Integration Definition for Function Modeling (IDEF0)	Входы и выходы 	Спецификации входов и выходов принципиально не отличаются и выполняются аналогично спецификациям потоков данных.
Unified Modeling Language: Activity Diagram	Потоки управления (переходы) 	Специфицируются только помеченные потоки управления, обозначающие передачу данных (документов).
Business Process Modeling Notation (BPMN)	Потоки сообщений 	Могут специфицироваться все потоки сообщений.

Заключение. По результатам проделанной работы можно сделать вывод, что проектные диаграммы на языке NVL связываются с диаграммами, моделирующими бизнес-процессы посредством информационных объектов взаимодействия. Эти объекты специфицируются с помощью потоков N-модели, которые представлены на проектных диаграммах в виде некоторых множеств атрибутов (см. таблицу).

Библиографический список

1. Ольховик О.В. N-модель данных / О.В. Ольховик, А.В. Белых // Изв. Южного федер. ун-та. Техн. науки. Темат. вып. «Интеллектуальные САПР». – 2009. – №8.
2. Белых А.В. Декларативный язык для N-модели данных / А.В. Белых, С.М. Ковалев, О.В. Ольховик // Вестн. Рост. гос. ун-та путей сообщения. – 2009. – №4.
3. Белых А.В. Визуально-декларативный язык для проектирования программного обеспечения информационных систем / А.В. Белых, С.М. Ковалев, О.В. Ольховик // Вестн. Донск. гос. техн. ун-та. – 2009. – №4.
4. Белых А.В. Сокращение избыточности схемы объектно-ориентированных баз данных / А.В. Белых, С.М. Ковалев, О.В. Ольховик // Изв. Южного федер. ун-та. Техн. науки. Темат. вып. «Интеллектуальные САПР». – 2009. – №12.
5. Белых А.В. Нормализация объектно-ориентированных баз данных на основе N-модели данных / А.В. Белых, С.М. Ковалев, О.В. Ольховик // Вестн. ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. – 2009. – Вып.7.
6. Federal Information Processing Standards Publication 184. Integration definition for function modeling (IDEF0) [Electronic resource]. – Mode of access: <http://www.idef.com/pdf/Idef0.pdf>
7. OMG. Unified Modeling Language: Superstructure, version 2.2. [Electronic resource]. – Mode of access: <http://www.omg.org/cgi-bin/doc?formal/09-02-02.pdf>

8. OMG. Business Process Modeling Notation (BPMN) Version 2.0 [Electronic resource]. – Mode of access: <http://www.omg.org/spec/BPMN/2.0/Beta2/PDF/>

Материал поступил в редакцию 15.11.10.

References

1. Ol'hovik O.V. N-model' dannyh / O.V. Ol'hovik, A.V. Belyh // *Izv. Yujnogo feder. un-ta. Tehn. nauki. Temat. vyp. «Intellectual'nye SAPR»*. – 2009. – №8. – In Russian.

2. Belyh A.V. Deklarativnyi yazyk dlya N-modeli dannyh / A.V. Belyh, S.M. Kovalev, O.V. Ol'hovik // *Vestn. Rost. gos. un-ta putei soobscheniya*. – 2009. – №4. – In Russian.

3. Belyh A.V. Vizual'no-deklarativnyi yazyk dlya proektirovaniya programmnoho obespecheniya informacionnyh sistem / A.V. Belyh, S.M. Kovalev, O.V. Ol'hovik // *Vestn. Donsk. gos. tehn. un-ta*. – 2009. – №4. – In Russian.

4. Belyh A.V. Sokraschenie izbytochnosti shemy ob'ektno-orientirovannyh baz dannyh / A.V. Belyh, S.M. Kovalev, O.V. Ol'hovik // *Izv. Yujnogo feder. un-ta. Tehn. nauki. Temat. vyp. «Intellectual'nye SAPR»*. – 2009. – №12. – In Russian.

5. Belyh A.V. Normalizaciya ob'ektno-orientirovannyh baz dannyh na osnove N-modeli dannyh / A.V. Belyh, S.M. Kovalev, O.V. Ol'hovik // *Vestn. VolgGTU. Ser. Aktual'nye problemy upravleniya, vychislitel'noi tehniki i informatiki v tehniceskikh sistemah*. – 2009. – Vyp.7. – In Russian.

6. Federal Information Processing Standards Publication 184. Integration definition for function modeling (IDEF0) [Electronic resource]. – Mode of access: <http://www.idef.com/pdf/Idef0.pdf>

7. OMG. Unified Modeling Language: Superstructure, version 2.2. [Electronic resource]. – Mode of access: <http://www.omg.org/cgi-bin/doc?formal/09-02-02.pdf>

8. OMG. Business Process Modeling Notation (BPMN) Version 2.0 [Electronic resource]. – Mode of access: <http://www.omg.org/spec/BPMN/2.0/Beta2/PDF/>

O.V. OLKHOVIK

DESIGN CHARTS IN N-VISUAL LANGUAGE AND BUSINESS PROCESS MODELING

Problems that permit solution of charts in N-Visual Language dealing with design and automatic code generation of the software information systems are described. Collecting of design diagrams in N-Visual Language with modeling business processes diagrams in various notations is considered.

Key words: *business process modeling, information system, CASE-technologies, Model-driven engineering, visual-declarative language, software engineering.*

ОЛЬХОВИК Олег Владимирович (р. 1973), начальник отдела разработки и внедрения информационных технологий ВЦ Донского государственного технического университета, кандидат технических наук (2000). Окончил Ростовский государственный университет путей сообщения (1995). Область научных интересов: базы данных, информационные системы. Автор 14 научных публикаций.

olvick@spark-mail.ru

Oleg V. OLKHOVIK (1973), Head of the IT Development and Utilization Department, Information Computer Centre, Don State Technical University. Candidate of Science in Engineering (2000). He graduated from Rostov State Transport University (1995). Research interests: databases, information systems. Author of 14 publications.