

ПРЕОБРАЗОВАНИЕ КОНФИГУРАЦИИ «1С:ПРЕДПРИЯТИЯ» В UML-МОДЕЛЬ

А. С. Балюк, В. А. Попова

Многие современные программные системы имеют сложную архитектуру из-за непрерывного внедрения в них новой функциональности. Поэтому уместно использовать средства, которые повышают удобство создания программного обеспечения. Так, существуют механизмы, которые позволяют представлять архитектуру систем в графическом виде [1], чтобы было удобнее и быстрее отслеживать взаимосвязь объектов [2].

На текущий момент для представления структуры программ применяются диаграммы классов языка UML. Для многих языков программирования, в том числе для C++, Javascript, Java, Python, существуют механизмы, например, Visual Paradigm [3] и Star UML [4], которые предоставляют возможность преобразования кода программы в UML [5], а именно в диаграмму классов. Но в настоящее время не существует средств, позволяющих визуально представлять структуру прикладных решений на платформе «1С:Предприятие» [6]. В связи с этим возникла идея создания программы для конвертации конфигурации в UML-модель.

Любая конфигурация «1С:Предприятие» может быть выгружена в файлы XML-формата. Возможность выгрузки в XML позволяет преобразовать этот формат описания конфигурации в UML-модель, которая представляется в формате XMI (XML Metadata Interchange) [7].

Элементы диаграммы классов в XMI имеют следующие обозначения:

- `packagedElement` — определяет класс, содержащий атрибуты `xmi:id` (идентификатор класса) и `name` (наименование класса);
- `ownedAttribute` — обозначает атрибут класса и содержит свойства `xmi:id` (идентификатор атрибута), `name` (наименование атрибута) и `type` (тип атрибута);
- `ownedOperation` — описывает операцию класса, идентификатор и наименование которой записываются в атрибуты `xmi:id` и `name` соответственно; может содержать вложенные элементы `ownedAttribute`, обозначающие параметры операции (при значении атрибута `direction="in"`) и возвращаемые значения (`direction="return"`);
- `ownedMember` — отражает зависимость между классами, содержит атрибуты `xmi:id`, `client` (идентификатор зависимого класса), `supplier` (идентификатор класса-поставщика для зависимого класса);
- `generalization` — указывает, что класс, идентификатор которого записан в атрибуте `specific`, является наследником класса, указанного в `general`.

Структуру прикладных решений на базе «1С:Предприятие» составляют следующие типы метаданных:

1. «Общие объекты»: «Подсистема», «Общий модуль», «Роль» и т. д.

2. «Прикладные объекты»: «Справочник», «Документ», «Перечисление», «Регистр сведений», «План видов характеристик» и т. д.

Каждый тип метаданных обладает базовой функциональностью, которая определена в платформе «1С:Предприятие». При создании конфигурации разработчики создают общие и прикладные объекты, наследующие всю функциональность соответствующих типов метаданных, которую можно переопределить под потребности разрабатываемой системы.

Все объекты конфигурации имеют свойства, типами которых могут выступать прикладные объекты конфигурации и типы данных встроенного языка платформы «1С:Предприятие». Таким образом, для построения структуры конфигурации на базе «1С:Предприятие» требуется использовать информацию не только об объектах конфигурации, но и о типах встроенного языка.

Прямое преобразование XML-описания конфигурации 1С в UML обуславливалось сложной структурой XML-файлов, поскольку преобразование работает в обе стороны: то есть можно не только выгрузить конфигурацию в XML, но и восстановить её из XML, ввиду чего, помимо основных свойств, в XML-описании содержатся настройки поведения и отображения данных объектов конфигурации, которые не несут информации для графического представления структуры.

Поэтому было решено создать промежуточный формат, который был назван в рамках данной работы «Дерево типов конфигурации» (ДТК), а уже затем выполнить преобразование ДТК в XML.

Схема формата «Дерево типов конфигурации» и обозначения её элементов представлены на рис. 1.



Рисунок 1. XML-формат «Дерево типов конфигурации»

Для автоматического создания формата «Дерево типов конфигурации» была написана программа, которая:

1. Выполняет разбор исходных XML-файлов с данными конфигурации и HTML-файлов с данными о типах платформы при помощи запросов XPath.
2. Формирует XML-описание конфигурации в формате «Дерево типов конфигурации» с использованием механизма XDTO.

Преобразование конфигурации из ДТК в формат XMI сделано с помощью языка XSLT. Процесс конвертации из ДТК в XMI выполняется следующим образом:

1. Каждый элемент `context` соответствует классу в UML и поэтому определяется в XMI, как `packagedElement`.
2. Свойства элемента (`property`) преобразуются в `ownedAttribute`. У `property` может быть несколько типов (элементов `type`), а в нотации UML можно указать у атрибута только один тип. Поэтому, если у свойства имеется несколько типов, создаётся вспомогательный класс, в котором наименование определяется следующим образом: `name = СложныйТип<type(1), ..., type(k)>`, `k = Кол-во(type)`. Например, если в элементе `property` содержится два элемента `type`, обозначающие типы Число и Строка, то будет создан класс `СложныйТип<Число,Строка>`, который является типом для элемента `property`.
3. Все элементы `method` промежуточного формата преобразуются в UML в качестве операций класса. Поскольку в `method` перечислены всевозможные наборы параметров в виде `paramSet`, в XMI количество операций класса (`ownedOperation`) соответствует количеству элементов `paramSet`.
4. После формирования всех `packagedElement`, выполняется установка связей между классами. Сначала определяется, какие типы, без учёта повторений, есть в каждом классе у атрибутов, а также параметров и возвращаемых значений операций. Затем устанавливается отношение зависимости с каждым из классов, обозначающих типы атрибутов и операций. Если у элемента `context` присутствует атрибут `extends`, то класс `packagedElement` связывается с классом отношением наследования. Связь с табличными частями, а также с относящимися к нему формами, командами и макетами определяет отношение агрегации.

Для того чтобы выполнить тестирование разработанного программного комплекса, использовалась свободно распространяемая конфигурация «Библиотека стандартных подсистем».

На рис. 2 представлен фрагмент конфигурации в формате UML. Для визуализации применялась программа StarUML, в которую был импортирован файл XMI, описывающий структуру конфигурации.

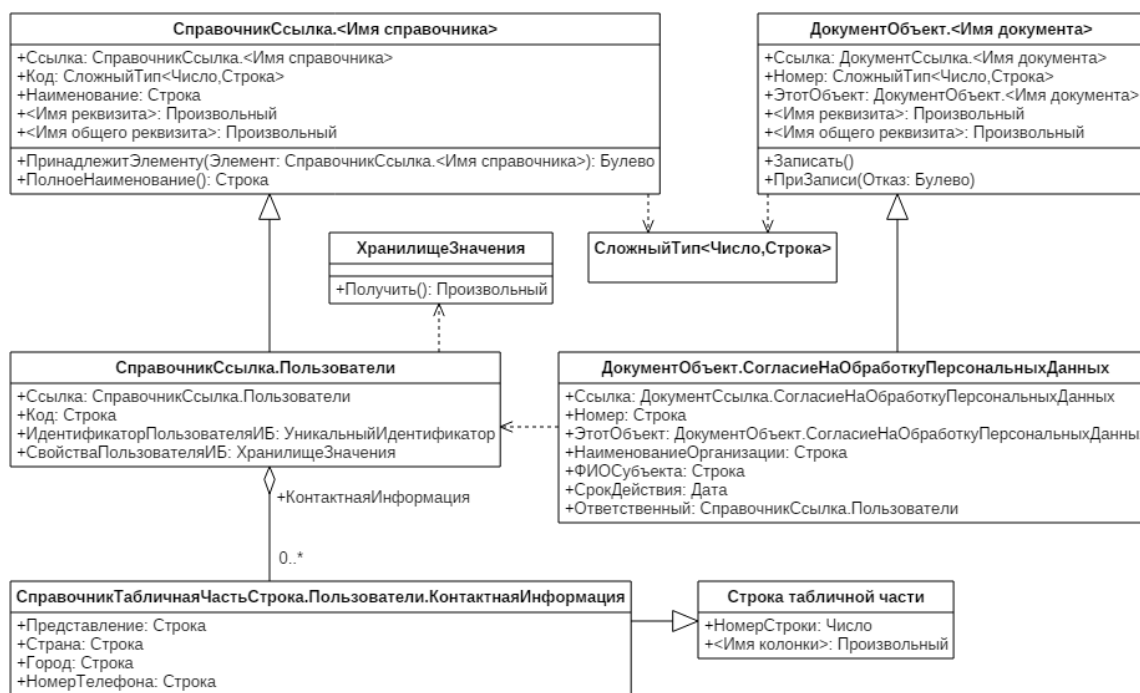


Рисунок 2. Представление фрагмента конфигурации «Библиотека стандартных подсистем»

В результате работы разработан программный комплекс, позволяющий выполнять преобразование любой конфигурации, разработанной на платформе «1С:Предприятие», в UML-модель.

Разработанный формат «Дерево типов конфигурации» также можно использовать в других задачах. Например, этот формат применяется в работе по созданию механизма статической проверки типов в 1С [8]: благодаря ДТК можно определять, какие типы значений может принимать та или иная конструкция встроенного языка платформы.

Литература

1. Романов, В. Ю. Моделирование свободно-распространяемого программного обеспечения с помощью языка UML / В. Ю. Романов // International journal of open information technologies. – 2013. – Т. 1. – № 27. – С. 11–15.
2. Ветеранова, Д. С. Описание структуры программного обеспечения в нотациях языка UML / Д. С. Ветеранова, В. С. Крылов, Э. А. Бекирова // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. – 2019. – № 26. – С. 66–72.
3. Visual Paradigm. Ideal Modeling & Diagramming Tool for Agile Team Collaboration [Электронный ресурс]. URL: <https://www.visual-paradigm.com/> (22.04.2021).
4. StarUML. A sophisticated software modeler for agile and concise modeling [Электронный ресурс]. URL: <https://staruml.io/> (22.04.2021).

5. About the Unified Modeling Language Specification Version 2.5.1 [Электронный ресурс]. URL: <https://www.omg.org/spec/UML/About-UML/> (23.04.2021).
6. 1С:Предприятие 8.3.17. Документация [Электронный ресурс]. URL: <https://its.1c.ru/db/v8317doc> (23.04.2021).
7. About the XML Metadata Interchange Specification Version 2.5.1. [Электронный ресурс]. URL: <https://www.omg.org/spec/XMI/About-XMI/> (23.04.2021).
8. Балюк, А. С. Статическая проверка типов для программ, разработанных на платформе «1С:Предприятие» / А. С. Балюк, В. А. Попова // Вестник Иркутского университета. – 2020. – № 23. – С. 54–55.