

УДК 681.51:(621.01+621.38)

Г. М. Мартинов, д-р техн. наук, проф.,

В. Л. Сосонкин, д-р техн. наук, проф.,

Московский государственный технологический университет "СТАНКИН"

Формализация данных STEP-NC-формата: фаза построения UML-модели

Предложен способ внедрения стандарта ISO 14649 STEP-NC, основанный на последовательной трансформации моделей и форматов данных в целях определения ключевого набора программных компонентов для работы с данными в стандарте STEP-NC. Рассмотрена ступень (фаза) трансформации моделей данных из EXPRESS в UML-представление.

Введение

Как стандарт формат ISO 14649 STEP-NC все еще находится в фазе активного развития. Сегодня готовы следующие версии частей стандарта: введение (Introduction), часть 10 (General Process Data, общие положения) и часть 11 (Milling, Drilling, фрезерование и сверление). Части 12 (Turning, токарная обработка) и 13 (Wide EDM, электроэрозионная обработка) находятся в процессе разработки, а часть 14 (Contour Cutting, обработка стекла, дерева, камня) пребывает в начальном состоянии. Включение в проект таких технологий, как быстрое прототипирование (Rapid Modeling), еще только предполагается.

Развитие STEP (Standart for the Exchange of Product Model Data) началось на рубеже 80—90-х годов прошлого века. Для описания форматов данных был предложен язык EXPRESS, а для графического представления (моделирования) — язык EXPRESS-G. STEP-NC унаследовал свойства этих языков.

В это же время происходили качественные изменения в области моделирования и представления данных. Так, появился графический язык моделирования UML (Unified Modeling Language) [1], ставший де-факто стандартом для визуализации, спецификации, конструирования и документирования систем, в которых ключевая роль принадлежит программному обеспечению (системы ЧПУ относятся именно к таким системам). В ноябре 1997 г. UML (версия 1.1) была принята в качестве стандарта Object Management Group (OMG) [2] под номером ISO 19501 [3] Согласно стандарту OMG, программные системы обмениваются UML-моделями

посредством формата XMI (XML Metadata Interchange).

Далее, появился стандарт XML (eXtensible Markup Language) разметки документов консорциума W3C [4]. XML определяет правила синтаксиса, используемые для разметки данных с помощью простых и доступных для чтения тегов.

Сегодня на фирмах, производящих системы ЧПУ, более 80 % инженеров связаны с разработкой программного обеспечения и лишь около 20 % занимаются аппаратурой. Это характеризует роль программного обеспечения в современной системе ЧПУ. На наш взгляд, бесполезно начинать внедрение формата STEP-NC в сложную программную систему без учета последних тенденций в области программирования.

Авторы на основе многолетнего опыта работы в научно-исследовательской лаборатории ЧПУ предлагают альтернативный путь внедрения STEP-NC, основанный на последовательной трансформации моделей и форматов данных, в целях определения формального набора компонентов для работы с форматом STEP-NC.

EXPRESS-модель формата STEP-NC дает иерархическое представление данных. EXPRESS-модель заложена в самом стандарте ISO 14649 и хорошо понятна лишь ограниченному кругу специалистов, занимающихся моделированием жизненного цикла изделий (Product Modelling). Современные программисты нуждаются в UML-модели формата STEP-NC, с помощью которой они будут разрабатывать редакторы, верификаторы и другие программные инструменты. UML-модель (рис. 1) представляет собой фазу трансформации, непосредственно следующую за EXPRESS-фазой. В последующих публикациях на базе UML-модели построим представление в виде электронного документа с файлами данных XML и файлами схем XSD (XML Schema definition language). Формат XML легко читается коммерческими анализаторами (parser) для построения объектной модели документа DOM (Document Object Model) и объектной модели схем SOM (Schema Object Model). Конечная цель состоит в определении формального набора программных компонентов, которые нужны производителям систем ЧПУ, станкостроителям и конечным пользователям для работы с управляющей программой в формате STEP-NC.

В этой статье построим UML-модель, опираясь на базовые понятия стандарта ISO 14649 STEP-NC [5] и используя некие формальные правила перехода из EXPRESS и EXPRESS-G в UML.

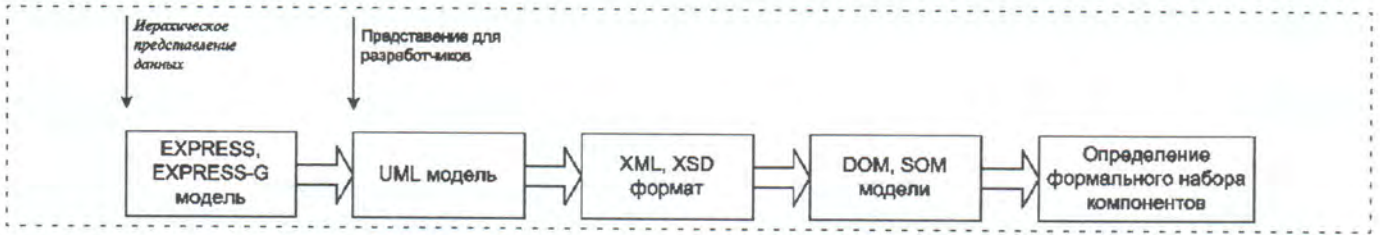


Рис. 1. Последовательная трансформация моделей и форматов данных. Фаза построения UML-модели

Представление об EXPRESS и EXPRESS-G

Международный стандарт STEP ISO-10303 специфицирует компьютерное представление и способы обмена информационными моделями изделия на протяжении его полного жизненного цикла. STEP использует язык EXPRESS (information model specification language, язык спецификации информационных моделей) и его графическую нотацию EXPRESS-G.

EXPRESS — это не язык программирования, поскольку не может быть откомпилирован или интерпретирован, но — язык согласования и формального описания информационных моделей STEP. В языке EXPRESS переплетены концепции объектно-ориентированного и процедурного подходов и концепция базы данных. Язык предоставляет полное и недвусмысленное описание статических моделей изделия в терминах *объектов данных* (entities), т. е. классов объектов, разделяющих общие свойства. Эти свойства представлены связанными *атрибутами и ограничениями* (constraints). В EXPRESS ограничения записываются с помощью элементов процедурного языка и деклараций.

Логически связанные *объекты данных* группируются в схемы. Информационная модель EXPRESS состоит из коллекции схем. *Схема* (Schema) в EXPRESS определяет мир понятий модели (Universe of Discourse, UoD). Каждая схема содержит:

- определения классов (entities), типов (types), функций (functions) и процедур (procedures);
- правила, определяющие зависимости между определениями;
- перекрестные зависимости: *использование* (USE specification) или *ссылки* (REFERENCE specification) между схемами.

Стандарт STEP-NC предлагает отдельную схему для каждой своей части. Например, схема части 10 (общие положения) выглядит так:

```
SCHEMA machining_schema;
(* Version 4 date: 2001-08-29
* Author: ISO TC184/SC1/WG7
*)
...
END_SCHEMA; (* machining_schema *)
```

Схема может содержать ссылки на объекты, определенные в других схемах. Например, схема для фрезерования (часть 11) ссылается на объекты данных, определенных из части 10: machine_functions, machining_operation, machining_tool:

```
SCHEMA milling_schema;
(* Version 18 date: 01-08-29
* Author: ISO TC184/SC1/WG7
*)
REFERENCE FROM machining_schema (*ISO14649-10*)
(...
machine_functions,
machining_operation,
machining_tool,
...
tool_direction);
...
END_SCHEMA; (*milling_schema *)
```

Здесь перечислены типы и объекты данных, на которые идет ссылка

Типы в языке EXPRESS могут быть элементарными, перечисляемыми и пользовательскими; все они применяются при построении объектов данных (entities) (рис. 2).

Элементарные типы Перечисляемые типы

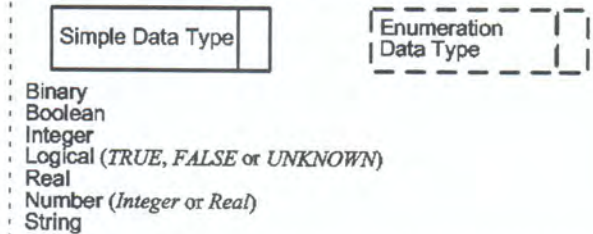


Рис. 2. Обозначения типов в языке EXPRESS

Элементарные типы (Simple Data Types) — это те, которые не могут быть разложены на другие типы. Графически элементарные типы обозначаются прямоугольником с двойной линией справа. *Элементарные типы*, предусмотренные в языке EXPRESS, представлены в табл. 1.

Следующий пример иллюстрирует использование типа BOOLEAN:

```
ENTITY toolpath
ABSTRACT SUPERTYPE OF (ONEOF (feedstop,
trajectory, parameterised_path));
its_priority : BOOLEAN;
```


Таблица 1
Таблица элементарных типов языка EXPRESS

Элементарные типы	Описание
REAL	Десятичное число. Десятичная точка должна присутствовать, даже если дробная часть равна нулю (например, 2.0).
INTEGER	Целое число.
NUMBER	Произвольное целое или действительное число.
LOGICAL	Значение может быть TRUE, FALSE или UNKNOWN.
BOOLEAN	Значение может быть только TRUE или FALSE.
BINARY	Последовательность битов, каждый из которых может быть установлен в 0 или 1.
STRING	Последовательность регистров зависящих символов.

```

its_type : toolpath_type;
its_speed : OPTIONAL toolpath_speedprofile;
its_technology : OPTIONAL technology;
its_machine_functions : OPTIONAL
machine_functions;
END_ENTITY;

```

Перечисляемый тип (Enumeration Data Type) предусматривает ряд возможных значений, определенных в списке. Только одно из этих значений может использоваться в экземпляре типа. Графически перечисление обозначается пунктирным прямоугольником с двойной линией справа (рис. 2). В STEP-NC перечисление используется, например, для определения направления поворота инструмента или направления кругового поворота детали.

```

TYPE rot_direction = ENUMERATION OF (cw, ccw);
END_TYPE;

```

Направление поворота может быть по часовой стрелке (clockwise, cw) или против часовой стрелки (counter-clockwise, ccw).

Пользователь может определять собственные типы (User-Defined Type) с помощью декларации TYPE. Пример *пользовательского типа* в STEP-NC: rvalue, используемый как класс числовой константы nc_constant или как класс переменной nc-variable.

```

TYPE rvalue = SELECT (nc_constant, nc-variable);
END_TYPE;

```

Декларация SELECT определяет коллекцию других допустимых типов. Экземпляр типа может использовать только одно значение коллекции.

Для работы с множествами EXPRESS предоставляет разного рода *коллекции*:

- ARRAY — коллекция фиксированного размера с упорядоченными элементами; обозначается A[1:?].
- BAG — неупорядоченная коллекция, допускающая повторение элементов; обозначается B[1:?].

- LIST — упорядоченная коллекция, не допускающая повторения элементов; обозначается L[1:?].
- SET — неупорядоченная коллекция, не допускающая повторения элементов; обозначается S[1:?].

Пример использования списка упорядоченных элементов приведен ниже:

```

ENTITY workplan
SUBTYPE OF (program_structure);
its_elements: LIST [1:?] OF executable;
its_channel: OPTIONAL channel;
its_setup: OPTIONAL setup;
its_effect: OPTIONAL in_process_geometry;
WHERE
WR1: SIZEOF (QUERY
(it<*its_elements(it=SELF))=0;
END_ENTITY;

```

EXPRESS-G служит формальной графической нотацией языка EXPRESS, которая делает его более выразительным. EXPRESS-G ориентирован на отображение таких статических компонентов, как объекты данных, атрибуты, декларации типов и иерархии наследования. Но в EXPRESS-G существует дополнительная возможность отображать функциональные компоненты, глобальные и локальные правила, а также алгоритмы.

В силу популярности языка UML и его мощной поддержки со стороны языков программирования и программных инструментов возникает проблема конвертирования EXPRESS-схем в UML.

Правила трансформации EXPRESS-G моделей формата STEP-NC в UML-модели

Задачу трансляции EXPRESS-G-модели в UML-модель нельзя посчитать тривиальной [6, 7]; целый раздел стандарта STEP (Part 25 EXPRESS/UML mapping for XMI) посвящен этой проблеме [8].

Исторически предпринимались многие попытки создать конвертеры из EXPRESS-G в UML, но интерес представляют лишь две из них. Первым является коммерческий продукт "Rational Rose EXPRESS Extension" шведской фирмы SoftLab AB. Продукт реализован в виде встраиваемого модуля (AddIns) для CASE-системы "Rational Rose 98" [9]. К сожалению, фирма не поддерживает продукт для последующих четырех версий "Rational Rose". Вторая попытка создания такого конвертора — это разработка университета "Kaiserslautern" в Германии [10].

Создание транслятора моделей в обобщенном виде (как это пытались сделать в упомянутых выше вариантах) невозможно без некоторых ограничений. Поэтому не удастся использовать существующие решения. Задача может быть решена, если сконцентрироваться на конкретной модели STEP-NC, а не на всех теоретических возможных моделях EXPRESS-G.

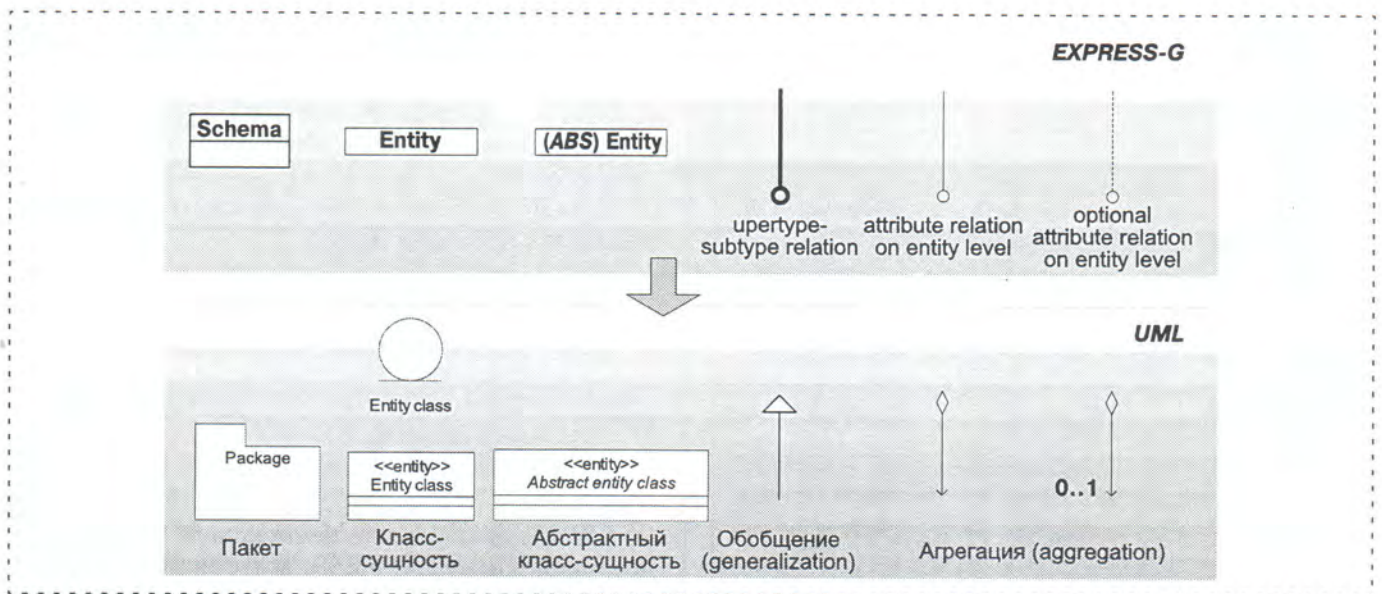


Рис. 3. Формальная замена элементов модели с одинаковой семантикой

EXPRESS-G не поддерживает динамические модели, но оперирует с данными статических моделей. Отсюда возникает первое правило.

Правило 1. Со стороны UML-представления достаточно использовать диаграммы классов, поскольку именно они определяют статическую структуру модели.

Некоторые базовые элементы модели имеют очевидное отображение. Например, объекты данных (entities) EXPRESS-G отображаются в классы, а схемы — в пакеты (packages), поскольку они разделяют одинаковую семантику.

Правило 2. Необходимо осуществлять формальную замену элементов модели, имеющих прямое семантическое соответствие.

В UML для моделирования данных и поведения с длинным жизненным циклом используют классы, обозначаемые как класс-сущность (entity class). Эти классы часто называют классами предметной области, поскольку они выступают в качестве абстракций предметов реального мира. Класс-сущность задают на базе обычного класса с помощью стереотипа¹ <<entity>>. Класс-сущность в нотации UML наиболее полно соответствует понятию сущности (entity) в EXPRESS-G (рис. 3). Абстрактные сущности EXPRESS-G имеют префикс (ABS) и отображаются в абстрактные классы-сущности в нотации UML.

Отношение supertype-subtype (supertype-subtype relation) в EXPRESS-G представляется толстой линией с кружочком на конце subtype. Оно

¹ Стереотип (stereotype) — расширение словаря UML, позволяющее создавать новые виды строительных блоков, аналогичные существующим, но специфичные для конкретной задачи.

соответствует символу обобщение (generalization) в UML между родительским классом (superclass) и дочерним классом (subclass).

При построении модели следует учитывать как видимость (доступность) самих элементов, так и видимость отношений, в которые вступают эти элементы (табл. 2).

Элементарные типы языка EXPRESS-G (Binary, Boolean, Integer, Logical, Real, Number и String) могут быть напрямую отображены в базовые типы UML. Пользовательские типы EXPRESS-G (например, identifier, Date, String) отображаются в классы. Другой вариант состоит в декларировании этих типов как базовых типов UML.

Правило 3. Необходимо отображать элементарные типы EXPRESS-G в базовые типы UML, а пользовательские типы декларировать как базовые типы UML.

Коллекции EXPRESS-G (SET, LIST, ARRAY, BAG) целесообразно реализовать в виде шаблонов (параметрических классов), поскольку уже существуют готовые библиотеки, предлагающие аналогичные функциональности. Примерами таких биб-

Таблица 2
Специальные символы видимости в UML

Символ обозначения	Описание
+	Общедоступный (public). Общедоступное свойство или функция, видимые для других классов.
#	Защищенный (protected). Защищенное свойство или функция, видимые для содержащего их класса или для унаследованных классов.
-	Закрытый (private). Закрытое свойство или функция, видимые только для содержащего их класса.



Рис. 4. Ключевые понятия и зависимости STEP-NC

ISO 14649 и приведены на рис. 4. Корневым элементом в STEP-NC является *проект* (project), который устанавливает стартовую точку для главного *плана* (workplan). План содержит набор из одного или более *изделий* (workpiece). В зависимости от контекста понятие workpiece также может быть интерпретировано как деталь или заготовка.

План (workplan) содержит последовательность исполняемых *шагов операции* (workingstep). Каждый *шаг операции* ассоциирует *объект-переход* (operation) с *типовыми формами* (feature), привязанными к геометрии *изделия* (work-

piece). Шаг операции указывает, какую следует использовать стратегию и параметры обработки.

Базирование (setup) определяет ориентацию, способ установки и крепления заготовки (workpiece) на станке, другими словами, содержит информацию о координатной системе заготовки. План связан с изделием через базирование. Объект *базирование* содержит информацию о множестве заготовок или множестве установок одной заготовки.

Правило 4. Необходимо реализовывать коллекции на базе шаблонов готовых библиотек.

Применим правила формальной трансформации EXPRESS-G-модели формата STEP-NC для получения UML-прототипов компонентов функциональности (UOFs). На базе этих прототипов может быть построена обобщенная UML-модель

Переход из EXPRESS-G в UML позволяет оценить модель STEP-NC с точки зрения читаемости, прозрачности и меры сложности. Модель становится легко понятной разработчикам программного обеспечения.

Обобщенная UML-модель формата STEP-NC

Основные компоненты функциональности (Units of Functionality, UOFs) и ключевые зависимости STEP-NC специфицированы в 10-й части

Универсальная UML-модель формата STEP-NC

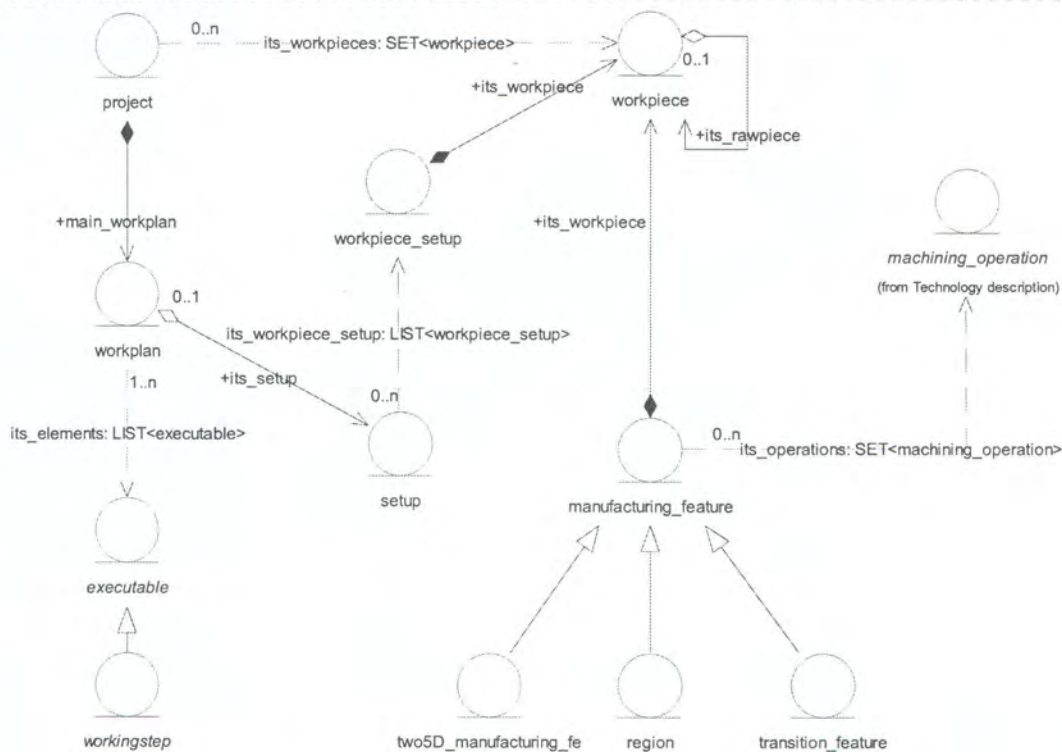


Рис. 5. Обобщенная UML-модель формата STEP-NC

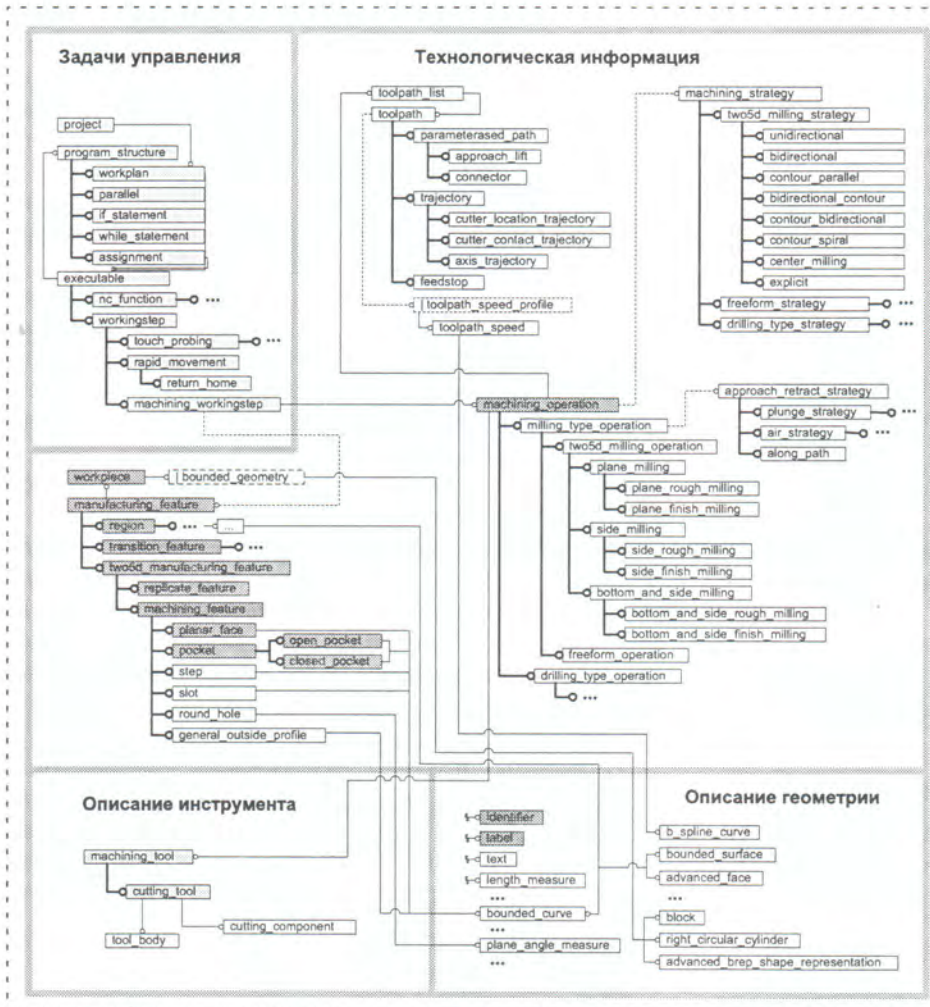


Рис. 6. Информация стандарта ISO 14649 на языке EXPRESS-G

формата STEP-NC, раскрывающая взаимоотношения на самом верхнем уровне абстракции.

Нетрудно найти на рис. 5 те же взаимосвязи, которые были показаны на рис. 4. Проект (project) содержит экземпляры рабочего плана (workplan) в своем m-поле (member field) плана main_workplan. Рабочий план (workplan) хранит шаги операции (workingstep) в упорядоченном списке (LIST) и т. д. В обобщенной UML-модели закрашены классы-сущности, соответствующие компонентам функциональности на рис. 4. Классы-сущности, чьи прототипы отсутствуют на рис. 4, остались незакрашенными. Обобщенная UML-модель формата STEP-NC отображает только те отношения, в которых находятся ключевые классы. Более полное представление о структуре классов и их более подробные взаимоотношения раскрывают другие классы диаграммы. Коллекции типа *неупорядоченный набор* (SET) и *упорядоченный список* (LIST) реализованы посредством классов шаблонов.

Практические рекомендации для построения UML-модели формата STEP-NC

В логическом представлении модели UML удобно объединить некоторое количество классов в группы (пакеты). *Пакет* (package) образует набор классов и других связанных пакетов. Такое объединение позволяет получить представление модели на более высоком уровне абстракции. Пакеты UML соответствуют разделам STEP-NC, которые также объединяют сущности логически.

На рис. 6 классы, использованные в этой статье при построении UML-модели, выделены цветом. Эти классы закреплены за следующими разделами стандарта ISO 14649:

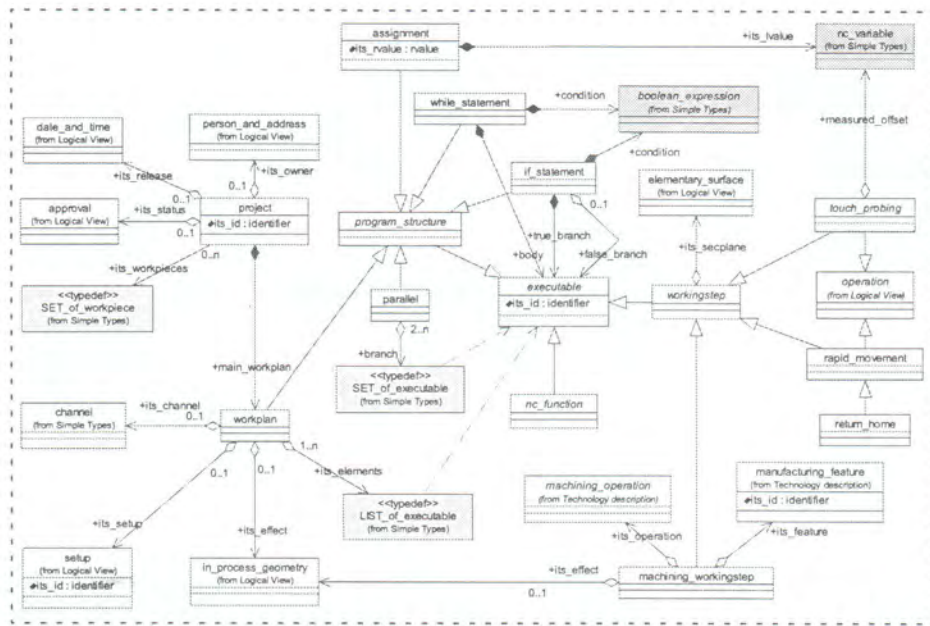


Рис. 7. Главная диаграмма классов формата STEP-NC

описания задач управления, технологической информации, описания инструмента и геометрического описания. Разделы, показанные на рис. 6, определяют основные пакеты UML-модели, представленной на рис. 7.

Пунктирными стрелками показано отношение зависимости между пакетами. Стрелка направлена от зависимого пакета-клиента (client package) к пакету-поставщику (supplier package).

Простые типы и типы, импортированные из других ISO 10303 схем, реализованы в пакете Simple Types. Пакет Simple Types — глобальный (global) для системы. Это означает, что объявленные в нем классы видны для остальных пакетов. Для выделения простых и импортированных типов в рассматриваемой UML-модели использован серый цвет.

На рис. 8 приведена диаграмма классов для раздела описания задач управления. Имена абстрактных классов, согласно UML-нотации, выделены курсивом (см. класс executable). Атрибуты классов, имеющие простые типы, организованы в виде m-полей (см. project::its_id). Остальные атрибуты реализованы с помощью агрегации. Набор SET для класса workplan построены с помощью конкретизации (instances) соответствующих классов-шаблонов (templates).

На диаграмме стереотип <<entity>> классов-сущностей опущен для избежания перегруженности модели. Назначение классов соответствует их прототипам из языка EXPRESS, описанным в [5] и [11].

На диаграмме классов "Обработка кармана" (Pocked class diagram) (рис. 9) показана модель технологического процесса обработки открытого и закрытого карманов (см. раздел технологической информации на рис. 6). Физические значения и их прототипы в UML-модели выделены красным и синим цветами для закрытого и открытого карманов, соответственно.

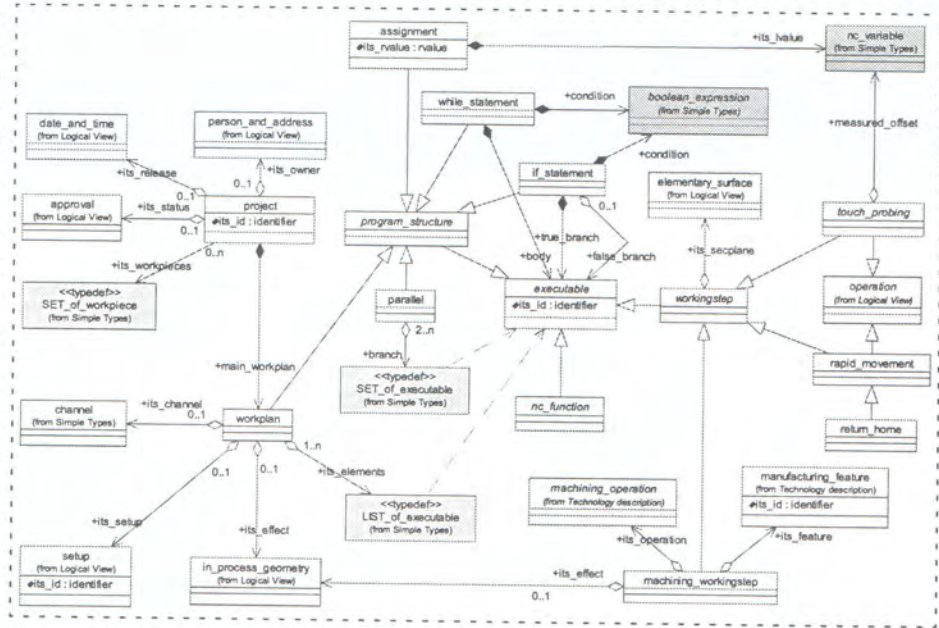


Рис. 8. Диаграмма классов раздела описания задач управления

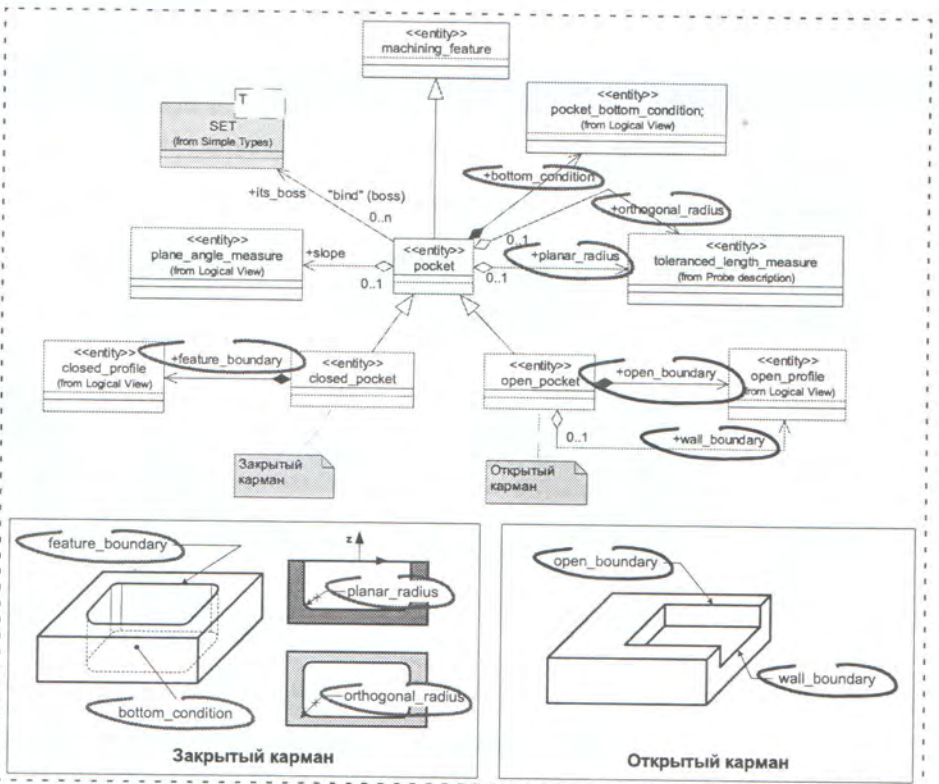


Рис. 9. Диаграмма классов "Обработка кармана"

На рис. 10 (см. четвертую сторону обложки) диаграмма классов обработки плоскости (Planar class diagram) описывает технологический процесс с помощью таких понятий, как заготовка (its_workpiece), обработанная поверхность (face_boundary), глубина снимаемого материала (depth), направление обработки (source_of_traveling) и т. д.

Заключение

В статье приведены только ключевые диаграммы UML-модели формата STEP-NC и соответственно изложены принципы их построения. Полная UML-модель может быть построена аналогично.

Наличие законченной UML-модели формата STEP-NC позволяет с помощью CASE-систем типа Rations Rose генерировать инструментарий, работающий с данными STEP-NC. Это могут быть редакторы данных, трансляторы, модули контроля синтаксиса, проверки геометрии и т. д.

Наглядность модели позволяет сформировать простые правила ее трансформации в XML-модель.

Список литературы

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Сер. "Для программистов": Пер. с англ. М.: ДМК Пресс, 2001. 432 с.
2. www.omg.org
3. ISO 19501-1 Unified Modeling Language (UML.)—Part:1 Specification.

4. Гарольд Э., Минс С. XML. Справочник. Пер. с англ. СПб: Символ-Плюс, 2002. 576 с.
5. Сосонкин В. Л., Мартинов Г. М. Понятийный аппарат комплекса производственных стандартов для числового программного управления оборудованием ISO 14649 STEP-NC (Standard for the Exchange of Product model data for NC) // Мехатроника, автоматизация, управление. 2004. С... (см. www.nc-systems.ru)
6. Hugh Davis. Bridging UML and STEP/EXPRESS with CDIF — Achieving Interoperability of Object and Product Data Models / In Jean Bezivin and Pierre-Alain Muller, editors. The Unified Modeling Language, UML'98 — Beyond the Notation. First International Workshop, Mulhouse, France, June 1998. P. 75—86.
7. Josh Lubell. From Model to Markup: XML Representation of Product Data // XML 2002 Conference, Baltimore, Maryland, 1 December 2002, www.mei.nist.gov/msidlibrary/doc/m2m.pdf
8. Technology reports. STEP/EXPRESS and XML, xml.coverpages.org
9. www.softiab.se
10. Florian Arnold and Gerd Podehl. Best of both worlds — a mapping from EXPRESS-G to UML / In Jean Bezivin and Pierre-Alain Muller, editors. The Unified Modeling Language, UML'98 — Beyond the Notation. First International Workshop, Mulhouse, France, June 1998. P. 45—54.
11. Сосонкин В. Л., Мартинов Г. М. Концепция числового программного управления мехатронными системами: интеграция на основе комплекса производственных стандартов STEP (Standard for the Exchange of Product model data) Информационные технологии в проектировании и производстве. М.: ВИМИ. 2003. № 2. С. 38—44. (см. www.ncsystems.ru).

CONTENTS

Kolesnikov A. A., Topchiev B. V. Synergetic Approach to a Problem of Controlled Systems Artificial Self-Organization Formation	2
Yudashkin A. A. A Synthesis of Self-Organizing Systems Capable to Memorize and Retrieve Multiple Own Configurations in a 3D-Space	7
Andriyanov A. I., Michalchenko G. Ya. Problems of Mathematical Modeling of the Closed Systems of Automatic Control on the Basis of Unipolar Reversive Modulation	11
Annenkov A. N. Asynchronous Actuating Motors with Heightened Power Parameters for Mechatronics Systems	19
Lipatov A. A., Sharovатов V. T. Executive Motor for the Force Cover Elements in Control System of the Pitch of the Helicopter	23
Vysotsky M. S., Belous M. M., Ivanov V. G., Liakhauv S. D. Active Safety System Design Experience for Heavy-duty Trucks (by the Example of ABS)	26
Yergin A. A., Kotiev G. O., Marokhin S. M. Development of Car Antislid Brake System	32
Guzij A. G., Bogomolov A. V., Kukushkin Ju. A. Fundamental Theory of a Functional — Adaptive Technique by Systems "Man—Machine" of a Heightened Accident Rate	39
Martinov G. M., Sosonkin V. L. Formalization of Data Step-NC of the Format: Phase of UML Construction of Model	49

Издательство «НОВЫЕ ТЕХНОЛОГИИ»

107076, Москва, Стромынский пер., 4/1

Телефон редакции журнала: (095) 269-5397, тел./факс 269-5510

Дизайнер Т.Н. Погорелова. Художник В.Н. Погорелов.
Технический редактор И.С. Павлова. Корректор Р.А. Чодарова

Сдано в набор 05.11.2004. Подписано в печать 15.12.2004. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 6,86. Уч.-изд. л. 7,96. Заказ 56. Цена договорная.

Журнал зарегистрирован в Комитете Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций
Свидетельство о регистрации ПИ № 77-11648 от 21.01.02

Учредитель: издательство "Новые технологии"

Отпечатано в Подольской типографии — филиале ОАО "ЧПК". 142110, г. Подольск, ул. Кирова, 15