

ИНЖИНИРИНГ ОНТОЛОГИЙ

УДК 004.9:658.512

DOI: 10.18287/2223-9537-2020-10-2-201-217

Особенности наследования информации в задачах интеграции систем технической подготовки производства**А.В. Щёкин***Мордовский государственный национальный исследовательский университет им. Н.П.Огарёва, Саранск, Россия***Аннотация**

Рассматриваются особенности наследования информации в технической подготовке производства при передаче данных из систем конструкторского проектирования (CAD) в системы технологической подготовки производства (CAM). К этим особенностям относятся объектно-ориентированный характер передаваемой информации и влияние PLM-контекстов на семантику инженерных данных. Объектно-ориентированный подход подразумевает наличие в процессах передачи информации «родителей» и их «потомков» и ассоциативных связей между ними, что делает возможным повторное использование проектных решений. UML-представление позволяет наглядно описать различные схемы наследования информации при интеграции CAM-систем с системами конструкторского проектирования. Эти схемы необходимо учитывать при реализации сквозных конструкторско-технологических проектов с высоким уровнем ассоциативности данных. Передача информации между подсистемами цифрового PLM-пространства предприятия происходит под влиянием информационных контекстов. Дано определение PLM-контекста как онтологии этапа жизненного цикла изделия, выступающего внешней информационной средой по отношению к интегрируемым приложениям. Разработан предварительный вариант онтологии предметной области, связанной с технологической подготовкой производства. Информацию об этапах жизненного цикла изделия предложено хранить непосредственно внутри 3D-модели объекта в форматах онтологического представления знаний на базе стандарта XML. Особенности наследования конструкторской информации рассматриваются на примере интеграции разрабатываемой CAM-системы с её базовой CAD-платформой КОМПАС-3D. Предложена стратегия автоматизации процесса влияния PLM-контекстов на значения передаваемых данных.

Ключевые слова: интеграция, техническая подготовка производства, CAD, CAM, CAPP, онтология, 3D-модель, КОМПАС-3D, XML, API.

Цитирование: Щёкин А.В. Особенности наследования информации в задачах интеграции систем технической подготовки производства / А.В. Щёкин // Онтология проектирования. – 2020. – Т. 10, № 2(36). – С.201-217. – DOI: 10.18287/2223-9537-2020-10-2-201-217.

Введение

Сложность семантической интеграции корпоративных приложений требует совместного применения принципов объектно-ориентированного моделирования, онтологического проектирования и современных подходов к формализации контекста. Технология объектного моделирования сегодня трактуется как универсальный способ представления знаний о предметной области (ПрО). «Ядро современных информационных технологий составляют объектные методы, искусственный интеллект и модели данных, причём рассматриваемые не по отдельности, а в неразрывной связи» [1]. В статье предпринята попытка объектно-

ориентированного описания механизма передачи данных от конструкторской 3D-модели к 3D-модели, используемой технологом при технологической подготовке производства.

Онтологии ПрО в задачах интеграции следует рассматривать не только как способ представления знаний об этапах жизненного цикла (ЖЦ) изделия, но и как глобальные интегрирующие модели данных (эталонные метаописания), обобщающие информацию, полученную от корпоративных приложений.

Аналогично тому, как семантика слов и предложений естественного языка зависят от контекста их применения, передача информации между подсистемами цифрового *PLM*-пространства (*Product Lifecycle Management* - прикладное программное обеспечение для управления ЖЦ продукции) предприятия происходит под воздействием внешнего информационного поля. Игнорирование этого факта приводит к потере или искажению актуальных значений передаваемых данных. Информационное поле предприятия представляет собой внешнюю среду по отношению к интегрируемым приложениям систем автоматизированного проектирования (САПР или *CAD*, *Computer-Aided Design*). Эта среда имеет иерархическую структуру, состоящую из вложенных друг в друга информационных контекстов, ближайшими из которых к задачам интеграции являются ПрО этапов ЖЦ изделия. При передаче данных между корпоративными информационными системами необходимо учитывать изменение значений инженерных данных под влиянием этих контекстов.

1 Проблемы интеграции корпоративной информации

Основные аспекты проблемы интеграции информационных систем рассмотрены в публикации [2]. К ним относятся архитектура системы интеграции, уровень взаимодействия данных (физический, логический, семантический), неоднородность источников информации, общая модель данных, способ представления интегрированных данных. Эти аспекты и связанные с ними задачи имеют непосредственное отношение к интеграции САПР и иных корпоративных приложений. При этом область САПР имеет некоторые особенности интеграции. Построение сквозных инженерных проектов предполагает создание между этапами ЖЦ изделия ассоциативных связей по типу «родитель-потомок». Сквозной проект — это когда изменения, внесённые в начальную стадию проекта, автоматически отображаются в последующих стадиях без дублирования информации в ручном режиме. Процессы обмена информацией между корпоративными приложениями следует рассматривать не просто как передачу данных, а как наследование информации, которое характеризуется наличием в этих процессах «родителей» и «потомков» и ассоциативных связей между ними, что делает возможным повторное использование проектных решений.

Наследование информации происходит под влиянием информационных контекстов. Например, при разработке технологического процесса необходимо учитывать текущее наличие режущих инструментов, оборудования, актуальные свойства материалов и иные факторы, информация о которых расположена за пределами среды технологического проектирования. Контекст в задачах интеграции — это всё, что не участвует непосредственно в процессе передачи информации между интегрируемыми приложениями, но влияет на значения передаваемых данных.

С учётом этих двух обстоятельств в процессе интеграции корпоративной информации можно выделить четыре стадии преобразования данных: считывание информации из приложения-источника, преобразование формата данных через эталонную модель, создание ассоциативных связей, корректировка значений данных с учётом влияния информационных контекстов. Этим стадиям можно сопоставить четыре проблемы интеграции (рисунок 1), расположив их на четырёх уровнях.

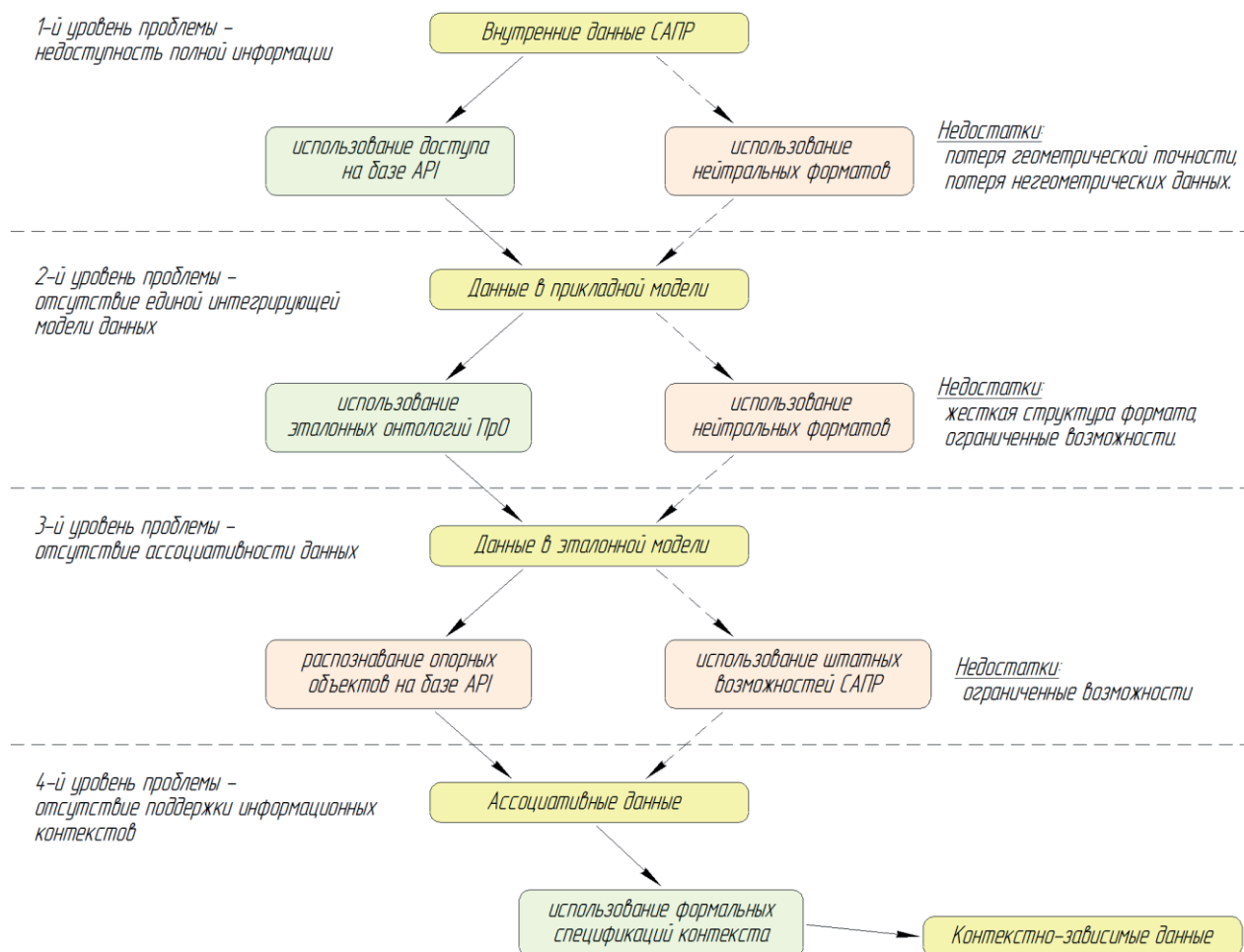


Рисунок 1 – Уровни проблемы интеграции корпоративных данных

Недоступность полной информации от приложений САПР является первым препятствием в реализации сквозных инженерных проектов. Под «полной информацией» здесь понимается исчерпывающий набор данных, необходимый для реализации корпоративной системы интеграции в условиях конкретного предприятия. Форматы коммерческих систем, как правило, являются закрытыми. Для обмена 3D-моделями САД-системы предлагают конвертацию данных через нейтральные форматы, наиболее развитым из которых является протокол STEP (STandard for Exchange of Product model data), применяемый в рамках технологии CALS (Continuous Acquisition and Lifecycle Support). Несмотря на развитие STEP, в обменные форматы выводится только геометрическая информация и, в лучшем случае, общие метаданные, например, наименование детали, разработчик, обозначение материала. Структура дерева построения модели, параметры формообразующих операций, свойства материала, как правило, остаются в статусе «секретной» информации, хотя соответствующие спецификации STEP для передачи этих данных давно разработаны. Причина такого положения кроется в рыночной конкуренции, под давлением которой разработчики САПР вынуждены создавать искусственные препятствия для интеграции собственных продуктов с продуктами других производителей САПР. Извлечь дополнительную информацию об изделии можно только с использованием API (Application Programming Interface) САД-системы и только в том объёме, который позволяет функциональность API. Извлекая данные через API, приходится работать с неоднородными источниками данных, связанными с 3D-моделью, например, свойства материала могут находиться в справочнике материалов, к которому требуется отдельный доступ

через *API* [3]. Существенный недостаток нейтральных форматов — это потеря точности геометрических данных, если приложения САПР работают на разных математических ядрах.

Второй уровень проблемы — это отсутствие единой интегрирующей модели данных. Глобальная модель данных в задачах интеграции всегда присутствует, но она не является универсальной. Использование нейтральных форматов в качестве интегрирующих моделей имеет ряд препятствий, обусловленных следующими причинами:

- игнорирование возможностей нейтральных форматов для передачи полной информации всеми разработчиками коммерческих САПР (обменные форматы используются на практике главным образом для передачи геометрии);
- нейтральные форматы имеют жёсткую структуру, фиксированные синтаксис и семантику, но всегда будут существовать объекты, процессы и наборы данных, не совпадающие полностью с актуальными спецификациями этих форматов;
- нейтральные форматы плохо предназначены для сжатия информации и передачи по сетям Интернет, а также визуализации в *HTML*-браузерах.

Перспективный путь решения данной проблемы — это разработка интеллектуальных конверторов данных, принимающих в качестве обменного протокола онтологические спецификации *Pro* на базе стандарта *XML (eXtensible Markup Language)*. Причём эти онтологии могут иметь различный уровень детализации *Pro* и различную таксономию понятий в зависимости от требований конкретной системы интеграции. Поскольку разнообразие корпоративных приложений весьма обширно, процесс разработки подобных конверторов должен быть автоматизирован с использованием новейших тенденций в области автоматизации программирования на основе онтологического подхода [4].

Ассоциативность данных является необходимым условием построения сквозных инженерных проектов, которая позволяет устранить многократное дублирование информации и обеспечить повторное использование проектных решений. Ассоциативность в САПР обычно реализуется с помощью специальных команд создания связанной геометрии, ссылок между переменными и свойствами объектов и часто используется в связке с параметризацией. Штатные возможности САПР по созданию ассоциативных объектов, как правило, ограничены или могут полностью отсутствовать в случае интеграции приложений, работающих на разных САПР-платформах. Автоматическое создание ассоциативных связей возможно на базе *API* интегрируемых приложений путём автоматического распознавания опорных (родительских) объектов.

Интерпретация любых форм информации зависит от контекста. Это утверждение справедливо не только для естественных языков как средства обмена информацией между людьми, но и любых способов передачи данных между техническими объектами. Знания, накопленные в области лингвистики, в полной мере применимы к искусственным языкам, в том числе к современным средствам межпрограммных взаимодействий.

Проблема семантики и контекста последовательно рассматривалась при анализе текстов в работах [5-9] и др. Современные исследователи выдвигают проблемы формализации контекста цифровой информации [10], потребности в интеграции и интероперабельности [11], обработки больших данных [12]. Простейшим способом формализации цифрового контекста является добавление к описанию предмета сопроводительной информации (метаданных). Более сложные подходы предлагают использование онтологий верхнего уровня, исчисления предикатов, теории математических категорий, дескрипционной логики и их расширений.

Любой контекст существует в другом, более общем контексте. В практических задачах необходимо в первую очередь обнаружить ближайшие контексты к рассматриваемому предмету. В статье рассмотрено влияние этапов ЖЦ изделия, соотнесенных с подсистемами *PLM*-

пространства предприятия, на интерпретацию инженерных данных при передаче информации от конструктора к технологу.

2 UML-представление наследования конструкторской информации

К разряду систем технологической подготовки производства относятся САПР ТП (*CAPP*-системы, *Computer-Aided Process Planning*) и системы автоматизированной разработки управляющих программ для станков с числовым программным управлением (ЧПУ) или *CAM*-системы (*Computer-Aided Manufacturing*). Эти системы в качестве исходных данных могут принимать как 2D-, так и 3D-информацию, полученную из систем конструкторского проектирования *CAD*. Но для целей построения сквозных конструкторско-технологических проектов целесообразнее использовать твёрдотельную модель, поскольку она несёт в себе не только полную информацию о геометрии объекта, но и допускает больше возможностей для включения в свой состав дополнительных метаданных.

Особенности наследования конструкторской информации можно представить на примере интеграции *CAM*-системы для платформы АСКОН с её базовым продуктом КОМПАС-3D. Автором статьи разрабатывается *CAM*-система с использованием среды программирования *Visual Studio C++*, *API* КОМПАС-3D и функций геометрического ядра *C3D* [13]. В настоящее время *CAM*-система включает в себя два модуля: *CAM*-приложение для токарной двухкоординатной обработки [14] и *CAM*-модуль для трёхосевого фрезерования [15]. *CAM*-система полностью интегрирована в пользовательский интерфейс КОМПАС-3D и использует в качестве источника конструкторской информации *CAD*-модель, созданную в КОМПАС-3D. Траектории обработки, которые генерирует *CAM*-система, ассоциативно связаны с опорными объектами обрабатываемой детали и автоматически перестраиваются при изменении положения и размеров элементов детали.

При моделировании технологического процесса рекомендуется работать не напрямую с конструкторской 3D-моделью, а с её ассоциативной копией, которая называется моделью технолога (рисунок 2). Это связано с тем, что технолог вынужден вносить изменения в 3D-модель обрабатываемой детали: необходимо поставить локальную систему координат, задающую направление осей станка; может потребоваться создать дополнительные построения на модели, например, сечение детали для удобства выбора внутренних поверхностей при программировании токарной обработки и др. Чтобы отделить изменения, сделанные технологом, и не загромождать конструкторскую модель технологическими данными, рекомендуется использовать копию исходной модели. Для построения сквозных конструкторско-технологических проектов 3D-модель технолога должна зависеть от модели конструктора, т.е. являться не просто копией файла конструкторской модели, а её ассоциативной копией. Ассоциативная связь в данном случае означает, что изменения, внесённые конструктором в собственную модель, автоматически передаются в модель технолога, т.е. модель конструктора выступает в качестве «родителя» для технологической модели. В общем случае модель технолога может зависеть от нескольких конструкторских моделей (например, при групповой обработке нескольких деталей на одной операции) и даже от параметров заготовки.

Удобным средством для описания подобных зависимостей является унифицированный язык моделирования *UML* (*Unified Modeling Language*), который позволяет построить наглядное графическое представление внутреннего состава класса и его отношений с другими классами.

Если рассматривать состав 3D-модели с точки зрения структуры данных, используемых для моделирования обработки в *CAM*-системе, то в списке атрибутов класса 3D-модели можно выделить пять элементов детали (рисунок 2):

- геометрию (*geometry*);
- параметрические переменные, объявляемые пользователем (*variables*);
- аннотации, т.е. обозначения размеров, допусков, шероховатостей (*annotation*);
- свойства материала (*material*);
- метаданные, т.е. сопроводительную текстовую и числовую информацию (*metadata*).

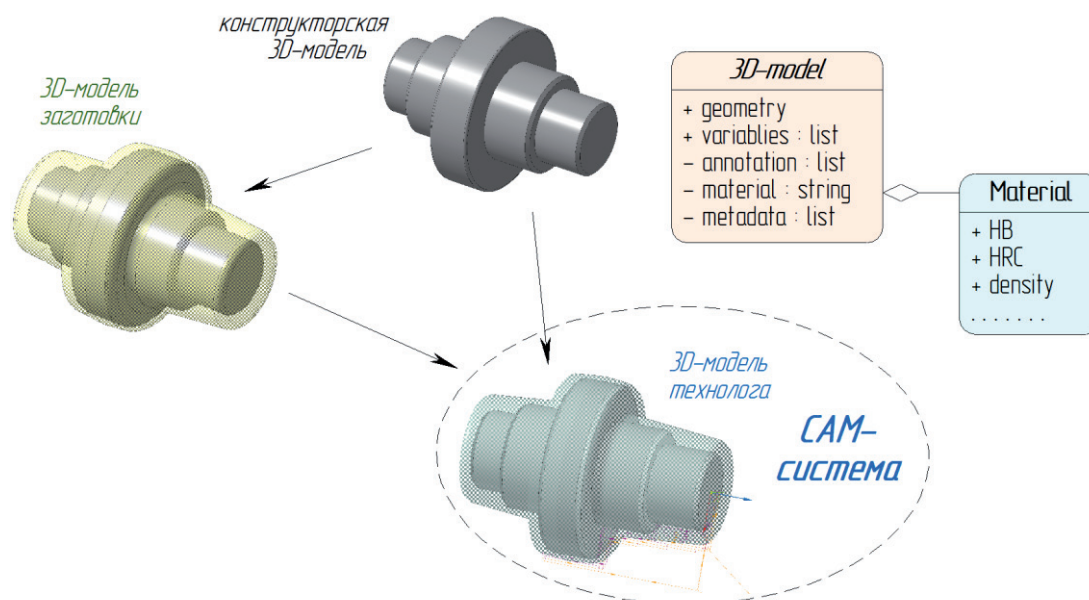


Рисунок 2 – Схема передачи информации от модели конструктора к модели технолога

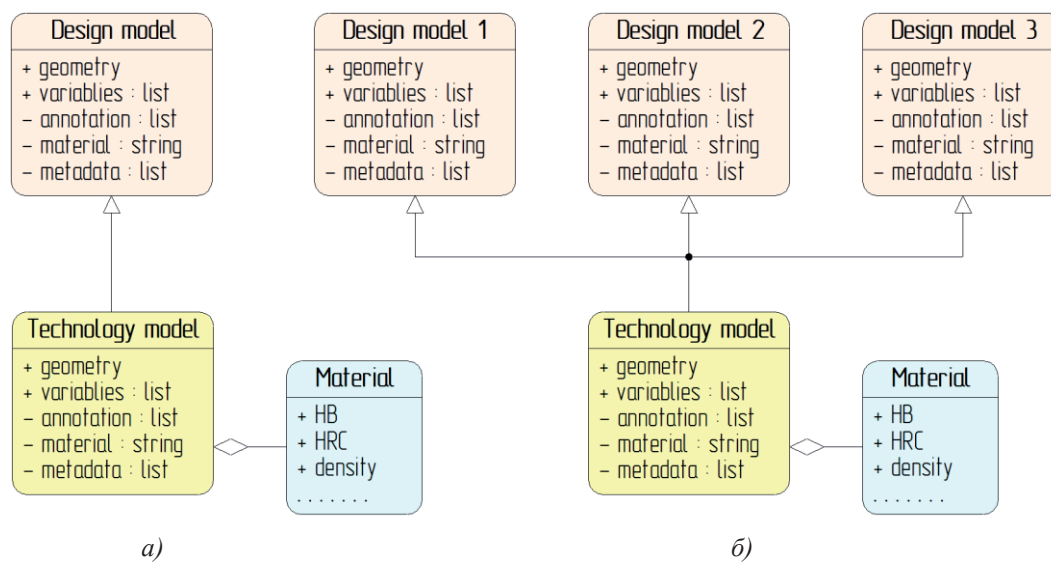
Геометрия модели представляет собой математическое описание поверхностей и структуру топологических связей между гранями, рёбрами и вершинами модели в граничном представлении *Brep* (*Boundary Representation*). Геометрия технологической 3D-модели может включать в себя два вида геометрии: ассоциативную геометрию, переданную из модели конструктора, и собственную геометрию, добавленную технологом (эта геометрия может отсутствовать, если технологу нет необходимости во вспомогательных построениях). Ассоциативная геометрия представляет собой полную копию *Brep*-представления конструкторской модели без истории её построения. Изменить её технолог не может, но может дополнить собственными построениями на технологической модели, например, сделать технологические отверстия с помощью операции вырезания или создать операцию прямого редактирования поверхностей. Ассоциативная геометрия и геометрия конструкторской модели находятся в отношении наследования, благодаря чему модель технолога обладает свойствами модели конструктора. Изменения, сделанные конструктором в геометрии изделия, автоматически передаются в ассоциативную геометрию технологической модели. Поскольку геометрия любой 3D-модели в системе КОМПАС-3D доступна в качестве ассоциативной геометрии для других моделей, то атрибут *geometry* следует считать открытым (публичным) членом класса *3D-model*.

Отличительной особенностью рассматриваемой *CAM*-системы является поддержка многоуровневой конструкторско-технологической параметризации [16], которая позволяет на стороне пользователей автоматизировать некоторые действия, например, расчёт режимов резания, создание собственных траекторий и упростить разработку постпроцессоров с помощью специальных макросов и параметрических формул. В рамках конструкторско-технологической параметризации можно использовать атрибуты *variables*, *annotation*, *material* и *metadata*. Извлечение негеометрической информации из технологической модели через *API* КОМПАС-3D подробно описано в статье [3]. Ни один из этих атрибутов не наследуется

из конструкторской модели через команду «Копировать объекты». Доступ открыт только к атрибуту *variables*, который рассматривается как открытый член класса *3D-model*, остальные считаются закрытыми (приватными) членами класса *3D-model*.

Атрибут *material* содержит строку с обозначением материала, а свойства материала хранятся в справочнике «Материалы и сортаменты» системы КОМПАС-3D. Поэтому свойства материала выносятся в отдельный класс, связанный с классом 3D-модели отношением агрегации. Аналогичным образом можно представить переменные, аннотации и метаданные отдельными классами, связанными с классом 3D-модели отношением композиции. Чтобы не загромождать диаграммы классов, эти атрибуты представлены упрощённо в виде списков в составе класса *3D-model*.

На рисунке 3 показаны два варианта наследования информации от конструктора к технологу. Как видно из этого рисунка штатными средствами КОМПАС-3D из конструкторской модели наследуются только два атрибута: *geometry* и *variables*. Аннотации, материал и метаданные технолог вынужден переопределять в собственной модели. Аннотации и метаданные могут не совпадать с исходной моделью (конструкторские обозначения могут быть излишни для технологической модели, а технолог может использовать свои аннотации), тогда материал, наименование и обозначение детали технологу приходится дублировать вручную. Открыть доступ ко всем элементам модели можно только через программный интерфейс приложения *API CAD*-системы, если реализовать в составе функций *CAM*-системы команду, аналогичную команде «Копировать объекты», но с расширенными возможностями передачи состава 3D-модели. При этом необходимо принимать во внимание возможные коллизии при наследовании информации от нескольких конструкторских моделей. Например, если исходные модели имеют разный материал, то в этом случае материал технологической модели должен назначить технолог.



а) – вариант с единственной исходной моделью; б) – вариант, когда технологическая 3D-модель наследует данные от двух и более конструкторских моделей (этот вариант соответствует групповому методу обработки)

Рисунок 3 – Варианты наследования данных от конструкторских моделей к модели технолога

В случае, когда в наследовании данных используется заготовка, то для повышения уровня автоматизации конструкторско-технологического проекта и его повторного использования 3D-модель заготовки может быть создана как параметризованная модель, ассоциативно связанная с геометрией и переменными конструкторской модели. На рисунке 4 показан пример параметризованной детали в форме диска (рисунок 4а) и её заготовка в виде штамповки

(рисунок 4б). В этом примере реализована схема наследования информации, приведённая на рисунке 2, когда и заготовка, и технологическая модель одновременно зависят от конструкторской модели. Заготовка содержит ссылочные переменные на размеры исходной детали и собственные переменные. Топология 3D-модели заготовки не является фиксированной, например, при достижении минимальной разницы между диаметрами ступеней детали выключается построение соответствующих ступеней заготовки. На рисунке 4б поверхности заготовки показаны прозрачными, сквозь которые видна ассоциативная геометрия, полученная от исходной детали. При изменении формы исходной модели перестраивается и модель технолога, и заготовка, а вслед за ними и ассоциативные траектории обработки, сгенерированные САМ-системой по технологической модели (рисунок 5). Модель заготовки САМ-система перестраивает автоматически через API КОМПАС-3D перед расчётом траекторий обработки.

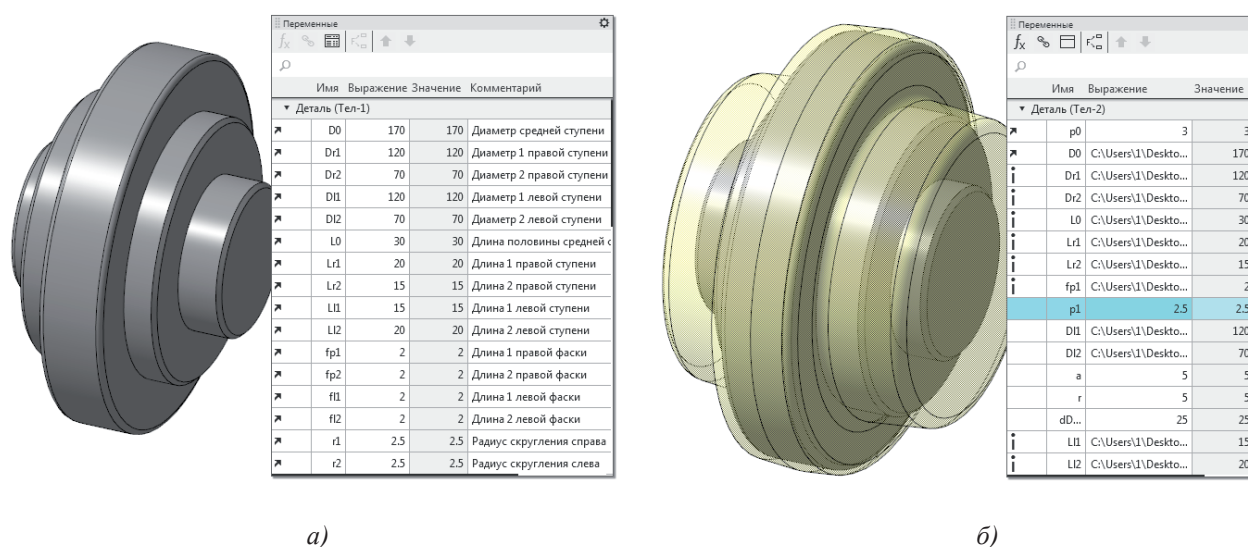


Рисунок 4 – Деталь в форме диска (а) и ассоциативно связанная с ней заготовка (б)

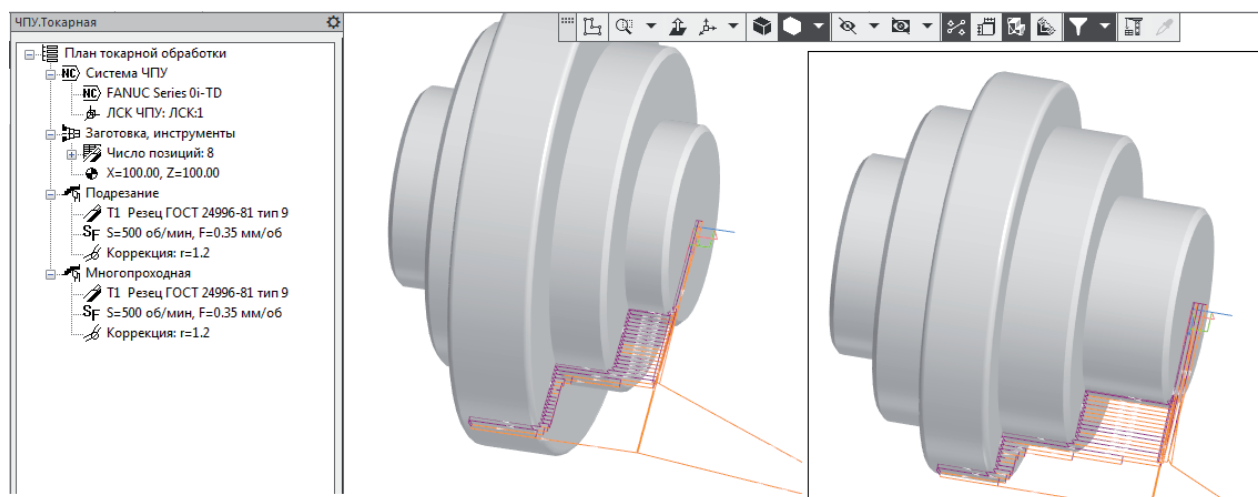
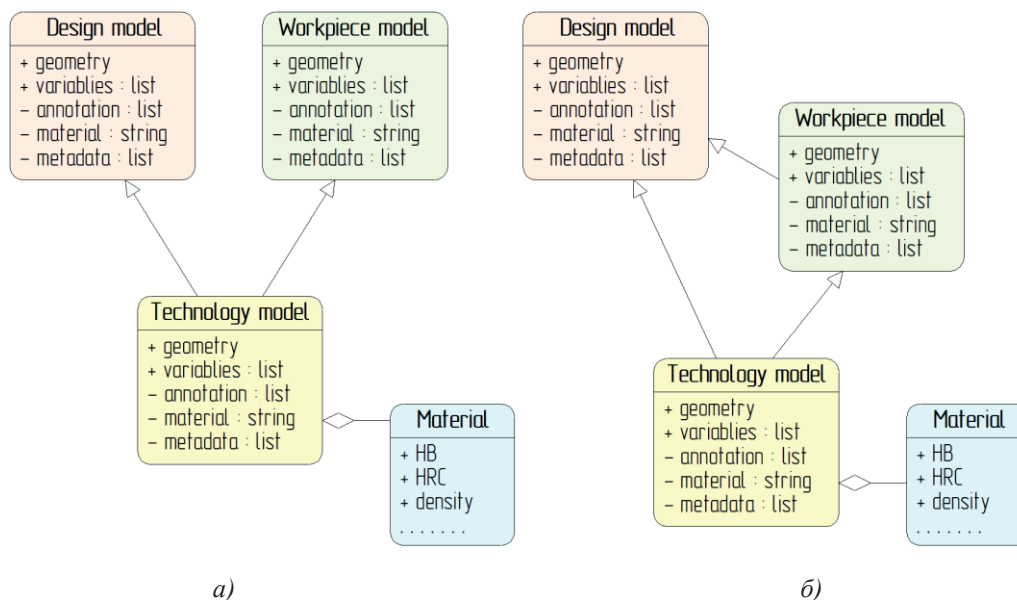


Рисунок 5 – Автоматическое перестроение ассоциативных траекторий обработки

Возможна ситуация, когда технологическая модель наследует данные одновременно и от конструкторской модели, и от заготовки. Обычно передавать геометрию от заготовки в технологическую модель нецелесообразно, потому что геометрия заготовки закрывает геометрию детали. Наследовать информацию из заготовки имеет смысл только в том случае, если для технологических расчётов требуется получить какие-либо переменные заготовки или пара-

метры аннотаций. На рисунке 6 показаны два варианта наследования данных с использованием модели заготовки. В каждом варианте вместо одной конструкторской модели может присутствовать несколько исходных моделей (такой случай является объединением рисунков 3б и 6).

На рисунке 7 приведён пример ромбовидного наследования, когда модель заготовки наследует данные конструкторской детали через промежуточные варианты заготовки.



- а) – вариант, когда модель заготовки не зависит от конструкторской модели;
- б) – вариант, когда модель заготовки зависит от САD-модели

Рисунок 6 – Варианты наследования данных с использованием модели заготовки

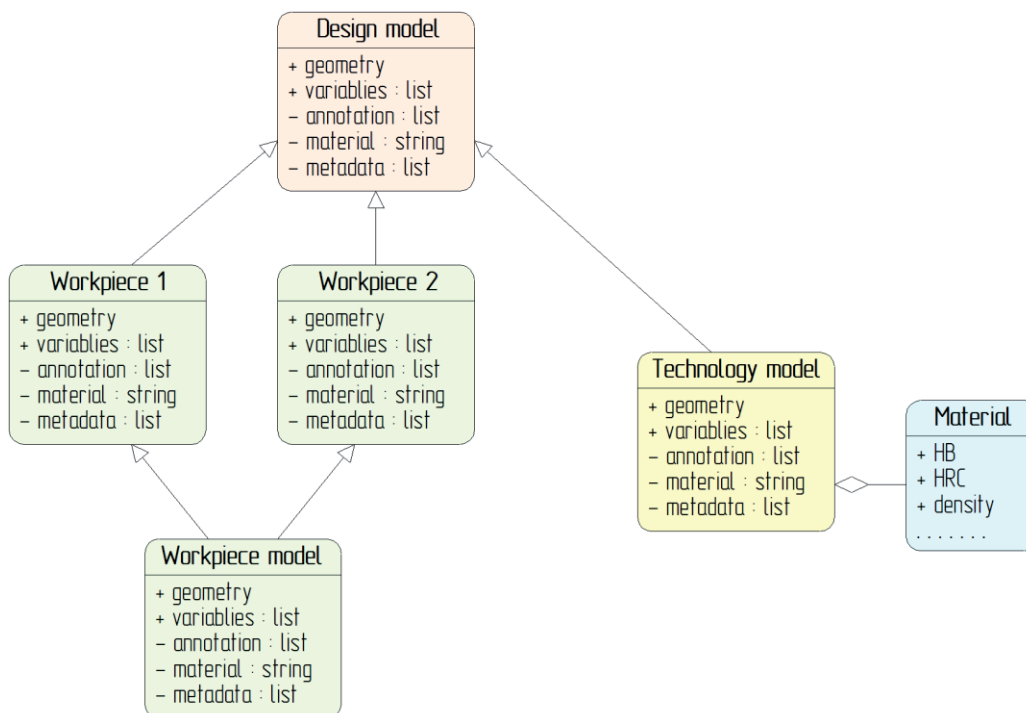


Рисунок 7 – Пример ромбовидного наследования данных от детали к заготовке

Модель штамповки, показанная на рисунке 4б, имеет плоскость разреза перпендикулярно оси детали. Путём изменения значений параметрических переменных деталь (рисунок 4а) может трансформироваться из диска в вал, у которого длина может в несколько раз превышать наибольший диаметр. Но для вала требуется другой вид штамповки, у которой плоскость разреза должна проходить через ось детали.

Таким образом, для данной детали требуется два варианта штамповки (модели заготовки). Алгоритмы их трёхмерного моделирования различны и должны быть реализованы в двух разных параметризованных 3D-моделях. Для создания сквозного конструкторско-технологического проекта необходимо иметь одну модель заготовки, подключенную к САМ-системе. Эта общая 3D-модель заготовки должна наследовать ассоциативную геометрию от обоих вариантов штамповки и содержать в своём дереве построения две операции копирования объектов, из которых только одна должна быть включена в расчёт в зависимости от соотношения габаритов детали. Такая схема представляет собой классический пример ромбовидного наследования (рисунок 7). Габаритные размеры детали могут быть унаследованы как от первого варианта, так и от второго варианта заготовки с помощью ссылочных переменных. В этих переменных указывается путь к файлу родительской 3D-модели, поэтому ссылки обеспечивают однозначный доступ к одноименным атрибутам класса 3D-модели.

UML-представление позволяет наглядно описать различные варианты наследования информации в задачах интеграции САМ-систем с системами конструкторского проектирования. Эти схемы наследования необходимо учитывать при реализации сквозных конструкторско-технологических проектов с высоким уровнем ассоциативности данных.

3 PLM-контекст цифрового пространства предприятия

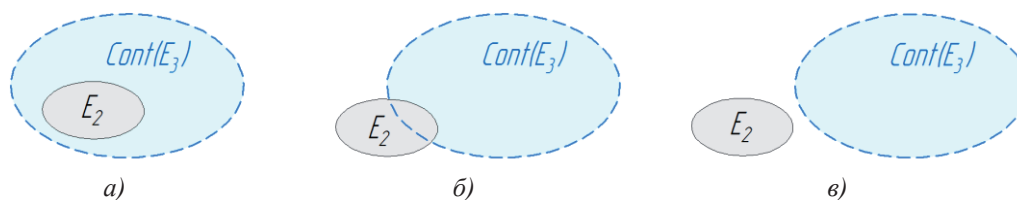
В рамках концепции Индустрия 5.0 ЖЦ изделия представляет собой пространство интеграции Интернета знаний и Интернета вещей. При этом этапы конструирования, технологической подготовки и планирования производства существуют в виртуальном мире, что делает возможным применение к ним Интернета знаний, а процессы изготовления продукта используют средства производства, рассматриваемые как вещи, что ведёт к возможности совместного применения Интернета знаний и Интернета вещей [17].

С точки зрения системного подхода любые этапы и стадии ЖЦ изделия следует рассматривать как элементы (*E*) его структуры. Элементы, взаимодействуя друг с другом путём обмена данными, находятся под влиянием всего информационного пространства предприятия и его неявных семантических полей. Например, конструктор, учитывая явные функциональные требования заказчика, изложенные в техническом задании, находится под влиянием собственного опыта, интуиции и культуры производства, принятой на данном предприятии. Технолог вынужден учитывать доступность оборудования, загруженного изготовлением других изделий, не связанных с его технологическим процессом. Инженер, занимающийся планированием производства, должен принимать во внимание загруженность складов с учётом сложных логистических связей предприятия с поставщиками и потребителями. Воздействие таких факторов, которые явно не специфицированы в рамках конкретного элемента ЖЦ изделия и находятся на метауровне относительно рассматриваемого процесса или явления, следует считать контекстным влиянием. Структура этого влияния сложная, трудно формализуема и обусловлена структурой конкретного предприятия. Эта структура имеет вложенность контекстов, например конструкторский контекст вложен в технологический, технологический — в организационный, некоторые контексты пересекаются между собой или дополняют друг друга.

Предлагается следующее определение *PLM*-контекста.

Определение: Пусть E_1 и E_2 — произвольные элементы ЖЦ изделия, участвующие в интеграции данных. Если при передаче информации от элемента E_1 к элементу E_2 данные меняют свои значения под влиянием третьего элемента E_3 , то онтология ПрО элемента E_3 , называется *PLM-контекстом* $Cont(E_3)$ для элемента E_2 .

ПрО *PLM-контекста* и элемента E_2 могут находиться в трёх отношениях множеств (см. рисунок 8).



а) – контекст содержит ПрО элемента $E_2 \in Cont(E_3)$; б) – ПрО пересекаются $E_2 \cap Cont(E_3)$;
в) – ПрО не пересекаются $E_2 \notin Cont(E_3)$

Рисунок 8 – Отношения ПрО элемента ЖЦ изделия и влияющего на него *PLM-контекста*

На рисунке 9 представлена схема передачи информации внутри фрагмента *PLM-пространства*, состоящего из четырёх элементов: конструкторское проектирование изделия (*CAD*), технологическая подготовка (*CAPP*), планирование и организация производства (*MES, Manufacturing Execution System*), технико-экономическое обоснование (*ERP, Enterprise Resource Planning*). Разработка операций для станков с ЧПУ в *CAM*-системах является стадией этапа технологической подготовки производства, т.е. $ПрО(CAM) \in ПрО(CAPP)$.

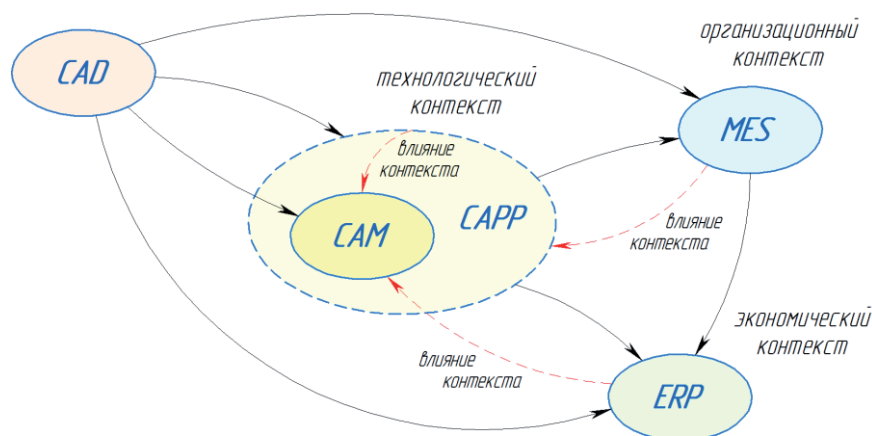


Рисунок 9 – Схема передачи информации внутри фрагмента *PLM-пространства*

Результаты конструкторского проектирования (*CAD*) используются практически всеми элементами ЖЦ изделия. Потенциально любой элемент ЖЦ может выступать в качестве контекста для любых других элементов *PLM-пространства* (на рисунке 9 потоки данных от контекстов обозначены штриховыми дугами). Поскольку в рамках данного исследования в первую очередь представляет интерес интеграция *CAM*-систем с системами *CAD*, то ниже приведены примеры контекстно-зависимой передачи данных для задач из области *CAM*.

Пример 1: учёт свойств обрабатываемых материалов.

В состав исходных данных для расчёта режимов резания входят свойства обрабатываемого материала. Например, такой параметр как твёрдость материала включён во многие методики расчёта технологических режимов. Если твёрдость материала брать из конструкторской 3D-модели, то это значение может быть неадекватным реальным свойствам материала

во время обработки. Конструктор назначает твёрдость исходя из функциональных требований, предъявляемых к изделию. Операции механической обработки часто производятся для материала в состоянии поставки. При передаче информации из конструкторской модели могут быть пропущены некоторые промежуточные операции технологического процесса. Узнать истинные свойства материала, можно, если обратиться к параметрам технологического процесса, в состав которого входит данная операция. В этом случае Про *CAPP*-системы выступает в качестве контекста по отношению к *CAM*-системе. Таким образом, свойства материала, извлечённые из *CAD*-модели, должны изменить свои значения под воздействием технологического контекста *CAPP*-системы. В свою очередь *CAPP*-система может находиться под влиянием организационного контекста *MES*-системы, например, если технолог вынужден учитывать загрузку оборудования. Это влияние *MES*-системы через технологический контекст передаётся и на *CAM*-систему.

Пример 2: автоматизированный подбор режущих инструментов.

Как правило, *CAM*-системы, содержат собственные базы инструментов. Эти базы могут не учитывать текущее наличие инструментов в цехе или возможность их поставки к моменту запуска изделия в производство. Для подбора инструментов необходима сверка инструментальной базы *CAM*-системы с базами *ERP*- или *MES*-систем. Такая сверка может быть произведена с использованием *MDM*-систем (*Material Data Manager*), предназначенных для совместного доступа и управления единой нормативно-справочной информацией [18].

4 Стратегия автоматизации процесса влияния *PLM*-контекстов на передаваемые данные

Предлагаемая стратегия автоматизации процесса влияния *PLM*-контекстов на значения передаваемых данных включает следующие этапы.

Во-первых, необходимо разработать подробные онтологии Про, связанных с этапами ЖЦ изделия. Эти онтологии должны выступать в качестве эталонных моделей интеграции, обладать свойствами расширяемости и адаптируемости в условиях конкретной системы интеграции. Необходимо выявить отношения множеств понятий этих онтологий: являются ли они иерархически вложенными, пересекаются или не имеют общих фрагментов. Предварительный вариант онтологии Про технологического процесса изготовления машиностроительной детали приведён на рисунке 10. Эта онтология отражает различные аспекты технологической подготовки производства. Детализация знаний о свойствах тех или иных понятий должна быть реализована отдельными онтологиями.

Во-вторых, необходимо проработать концепцию хранения информации об этапах ЖЦ изделия непосредственно в 3D-модели изделия. Существенным отличием предлагаемой концепции от известной методологии *CALS* является отказ от протокола *STEP* и замена его онтологическими спецификациями на базе *XML*-стандарта. Для хранения данных об этапах ЖЦ изделия можно использовать архивы 3D-моделей. Если открыть файлы деталей в форматах *SolidWorks* и КОМПАС-3D ZIP-архиватором, то можно увидеть содержимое этих архивов (рисунок 11).

Можно предположить, что геометрическая информация хранится внутри элемента данных *Contents* (это и есть атрибут *geometry* класса *3D-model* на рисунке 2). Анализ содержимого архивов показывает, что все файлы внутри 3D-модели *SolidWorks* являются бинарными, а внутри модели КОМПАС-3D элемент данных *MetaProductInfo* представляет собой текстовый *XML*-файл, содержащий общую информацию о составе изделия.

Существенным преимуществом такого подхода является то, что получение информации об этапе ЖЦ изделия может происходить из модели детали без необходимости обращения к

PDM- или MDM-системам. Хранение знаний об этапах ЖЦ изделия внутри 3D-модели является необходимым условием реализации многоагентных систем проектирования и управления производством [19]. Выступая в качестве интеллектуального агента среды проектирования и обладая при этом свойствами реактивности и целенаправленности, деталь должна все свои знания содержать внутри себя. Предварительный вариант онтологического содержания интеллектуального агента (детали) представлен на рисунке 12.

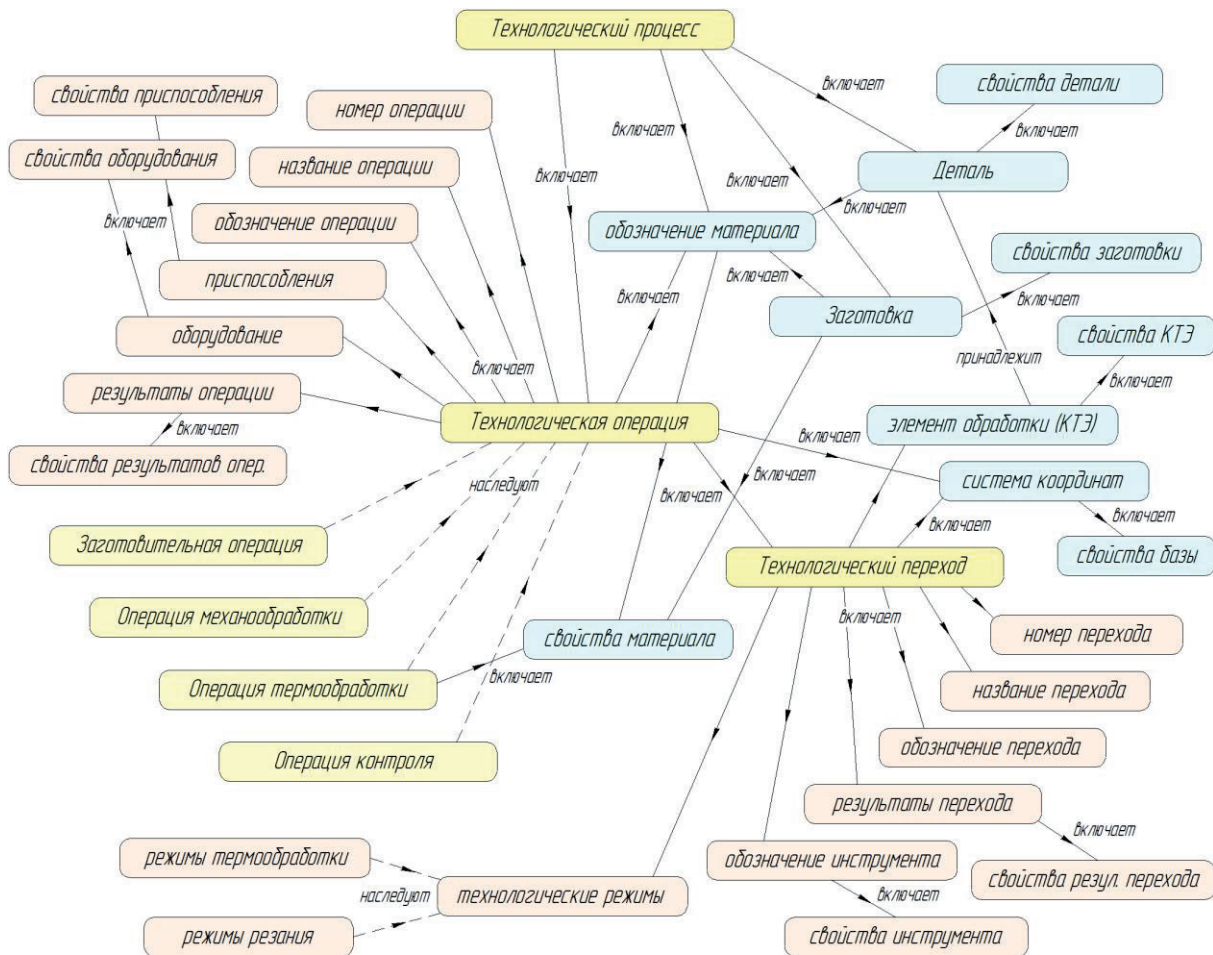


Рисунок 10 – Онтология Про технологического процесса

Имя	Размер	Сжатый	Файлов
Contents	44 498	45 504	11
SwDocMgrTempStorage	0	0	0
swXmlContents	768	832	2
ThirdPty	198	256	4
ThirdPtyStore	457	512	2
_DL_VERSION_4100	6	64	1
_MO_VERSION_4100	1 656	1 728	2
Config-0-Properties	164	192	
Header2	1 924	1 984	

а)

Имя	Размер	Сжатый	Версия
Contents	420 081	420 081	0
FileInfo	602	602	0
MetaInfo	214	120	0
MetaProductInfo	26 690	2 917	0
Preview	63 450	63 450	0
SysInfo	3 508	3 508	0

б)

Рисунок 11 – Содержимое 3D-модели в форматах SolidWorks (а) и КОМПАС-3D (б)

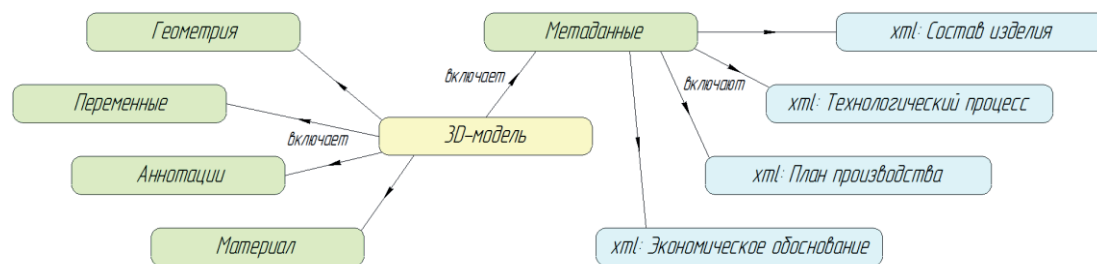


Рисунок 12 – Онтология знаний интеллектуального агента (детали)

В-третьих, необходима программная реализация интеллектуальных конверторов данных, принимающих в качестве интегрирующих моделей данных онтологии элементов ЖЦ изделия на базе стандарта *XML*. Разработка архитектуры таких конверторов должна быть основана на современных методах автоматизации программирования с применением *UML*-представлений объектов и онтологического подхода к разработке программного обеспечения.

Заключение

Особенности наследования инженерных данных необходимо учитывать при разработке САПР-приложений и создании на их основе систем интеграции корпоративного уровня. Приведено наглядное описание различных схем наследования информации с использованием *UML*-представлений на примере интеграции систем технической подготовки производства.

Обмен информацией между подсистемами предприятия происходит под влиянием информационных контекстов, обусловленных цифровым пространством предприятия. Ближайшими из этих контекстов к задачам интеграции являются Про ЖЦ изделия, обозначенные в статье как *PLM*-контексты. Для автоматизации процесса влияния *PLM*-контекстов на значения передаваемых данных необходима разработка онтологий Про, связанных с этапами ЖЦ изделия, и реализация на их основе интеллектуальных конверторов экспорта/импорта данных для приложений САПР.

Список источников

- [1] Грехэм, И. Объектно-ориентированные методы. Принципы и практика / И.Грехэм. – М.: Изд-во «Вилльямс», 2004. – 880 с.
- [2] Коголовский, М.Р. Методы интеграции данных в информационных системах / М.Р. Коголовский // Институт проблем рынка РАН, М.: 2010. – 9 с. - <http://www.ipr-ras.ru/articles/kogalov10-05.pdf>.
- [3] Щёкин, А.В. Автоматизация получения параметров детали для задач конструкторско-технологической параметризации / А.В. Щёкин // Инженерные технологии и системы. – 2019. – Т.29, №3. – С.345–365. DOI: 10.15507/2658-4123.029.201903.345-365.
- [4] Хорошевский, В.Ф. Проектирование систем программного обеспечения под управлением онтологий: модели, методы, реализации / В.Ф. Хорошевский // Онтология проектирования. – 2019. – Т. 9, №4(34). – С.429 –448. – DOI: 10.18287/2223-9537-2019-9-4-429-448.
- [5] Шлейермахер, Ф. О разных методах перевода // Вестник МГУ. Сер. 9: Филология. 2000. № 2. С. 127—145.
- [6] Гадамер, Х.-Г. Истина и метод: Основы философской герменевтики / Пер. с нем.; общ. ред. и вступ. ст. Б.Н. Бессонова. – М.: Прогресс, 1988. – 704 с.
- [7] Фреге, Г. Смысл и денотат // Семиотика и информатика. Вып. 8. М., 1977. С.181–210.
- [8] Carnap R., Bar-Hillel Y. An outline of theory of semantic information // Techn. Report of Res. Lab. Electr. – 1952. No. 247, 49 p.
- [9] Витгенштейн, Л. Философские исследования / Л. Витгенштейн // Философские работы. Ч. 1; пер. с нем. – М.: Гнозис, 1994. – С.75–320.

- [10] Коммюнике онтологического саммита 2018 — Контекст в контексте / K. Baclawski, M. Bennett, G. Berg-Cross, C. Casanave, D. Fritzsche, J. Luciano, T. Schneider, R. Sharma, J. Singer, J. Sowa, R.D. Sriram, A. Westerinen, D. Whitten. Пер. с англ. М.Д. Коровина, с сокр. // *Онтология проектирования*. – 2018. – Т.8, №2(28). – С.305-316. DOI: 10.3233/AO-180200.
- [11] Fritzsche, D., Gruninger, M., Baclawski, K., Bennett, M., Berg-Cross, G., Obrst, L., ... Westerinen, A. *Ontology Summit 2016 Communique: Framing the Conversation: Ontologies within Semantic Interoperability Ecosystems*. *Applied Ontology*, 2017; 12(2): 91–111. DOI: 10.3233/AO-170181.
- [12] Gruninger, M., Obrst, L., Baclawski, K., Bennett, M., Brickley, D., Berg-Cross, G., . . . Yim, P. *Ontology Summit 2014 Communique: The Semantic Web and Big Data meet Applied Ontology*. *Applied Ontology*, 2014; 9(2): 155–170. DOI: 10.3233/AO-140135.
- [13] **Камнев, А.** Интерфейс прикладного программирования геометрического ядра С3D, его применение и главное отличие от API системы КОМПАС-3D / А. Камнев // САПР и графика. 2016. №5. С.36–38. - <https://sapr.ru/article/25210>.
- [14] Модуль ЧПУ. Токарная обработка. - <https://kompas.ru/kompas-3d/application/machinery/module-chpu>.
- [15] Модуль ЧПУ. Фрезерная обработка. - <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo>.
- [16] **Щёкин, А.В.** Конструкторско-технологическая параметризация в составе интегрированной САМ-системы / А.В. Щёкин // Информационные технологии. 2019. №7. с. 34–54.
- [17] **Евгеньев, Г.Б.** Индустрия 5.0 как интеграция Интернета знаний и Интернета вещей / Г.Б. Евгеньев // Онтология проектирования. – 2019. – Т.9, №1(31). – С.7-23. – DOI: 10.18287/2223-9537-2019-9-1-7-23.
- [18] **Андриченко, А.Н.** Тенденции и состояние в области управления справочными данными в машиностроении / А.Н. Андриченко // Онтология проектирования. 2012, 2(4). – С. 25-35.
- [19] **Евгеньев, Г.Б.** Многоагентные системы полуавтоматического проектирования в машиностроении на базе механизма объект-функции / Г.Б. Евгеньев // Онтология проектирования. – 2020. – Т.10, №1(35). – С.50-62. DOI: 10.18287/2223-9537-2020-10-1-50-62.

Сведения об авторе



Щёкин Александр Васильевич, 1976 г. рождения. Окончил Мордовский государственный университет им. Н.П. Огарёва в 1998 г. Заведующий научно-исследовательской лабораторией «Автоматизация программирования станков с ЧПУ» ФГБОУ ВО «МГУ им. Н. П. Огарёва». Область научных интересов: САПР, автоматизация программирования станков с ЧПУ, САМ (Computer-Aided Manufacturing), онтологический инжиниринг. ResearcherID: F-4689-2019, ORCID: <https://orcid.org/0000-0001-5209-166X>, schekin@inbox.ru.

Поступила в редакцию 05.05.2020, после рецензирования 24.05.2020. Принята к публикации 08.06.2020.

The specifics of information inheritance in CAD/CAM-integration

A.V. Shchekin

National Research Mordovia State University, Saransk, Russia

Abstract

The article describes the specifics of information inheritance when transferring data from CAD system to CAM system: the object-oriented nature of the transmitted data and the influence of PLM contexts on data semantics. The object-oriented approach implies in the information transfer processes the presence of “parents” and “descendants” and associative relations between them, which makes it possible to reuse engineering projects. UML representation clearly describes the various schemes of information inheritance in the CAD/CAM-integration. These schemes must be taken into account when implementing end-to-end engineering projects. The PLM context is considered as the semantic field of the product life cycle stage, which affects the interpretation of engineering data. Digital space of an enterprise consists of informational contexts which affect the values of the transmitted data. Ignoring this circumstance leads to the loss or distortion of the actual values of the inherited parameters. The article provides a definition of the PLM context as an ontology of the product life cycle stage, which acts as the external environment in relation to integrated applications. A preliminary version of the ontology of the technological domain has been developed. It is proposed to store information about the stages of the product life cycle directly inside the 3D model of the product in the ontological representation of knowledge based on the XML standard. The specifics of information inheritance are considered on the example of integration of the CAM-system developed by the author of the article with its basic CAD-platform KOMPAS-3D. A strategy for automating the influence of PLM contexts on the values of the transmitted data is proposed.

Key words: integration, technical preparation of production, CAD, CAM, CAPP, ontology, 3D-model, KOMPAS-3D, XML, API.

Citation: Shchekin AV. The specifics of information inheritance in CAD/CAM-integration [In Russian]. *Ontology of designing*. 2020; 10(2): 201-217. – DOI: 10.18287/2223-9537-2020-10-2-201-217.

List of figures

- Figure 1 – Problem levels of enterprise data integration
- Figure 2 – The scheme of information transfer from the designer model to the technological model
- Figure 3 – Diagram of data inheritance from design models to the technological model
- Figure 4 – A disk-shaped part (a) and a workpiece associated with it (b)
- Figure 5 – Automatic regeneration of associative processing paths
- Figure 6 – Data inheritance diagram using workpiece model
- Figure 7 – An example of rhomboid data inheritance from a part to a workpiece
- Figure 8 – Relationships of the element of PLM and the PLM-context affecting it
- Figure 9 – Scheme of information transfer inside a fragment of PLM space
- Figure 10 – Ontology of the domain of technological process
- Figure 11 – Content of the 3D model in SolidWorks (a) and KOMPAS-3D formats (b)
- Figure 12 – Intelligent Agent Ontology of knowledge (details)

References

- [1] **Graham I.** Object-oriented methods. Principles and Practice. 3rd Edition. Addison-Wesley, 2001. 853 p.
- [2] **Kogalovskij MR.** Data integration methods in information systems [In Russian]. *Metody integracii dannyh v informacionnyh sistemah*. Institut problem rynka RAN, 2010. 9 p. <http://www.ipr-ras.ru/articles/kogalov10-05.pdf>.
- [3] **Shchekin AV.** Automation of Obtaining the Detail Parameters for Tasks of Design-Technological Parametrization [In Russian]. *Inzhenernyye tekhnologii i sistemy* = Engineering Technologies and Systems. 2019; 29(3): 345-365. DOI: 10.15507/2658-4123.029.201903.345-365.
- [4] **Khoroshevsky VF.** Ontology Driven Software Engineering: Models, Methods, Implementations [In Russian]. *Ontology of designing*. 2019; 9(4): 429-448. – DOI: 10.18287/2223-9537-2019-9-4-429-448.
- [5] **Shleiermakher F.** Different methods of translation. *Vestnik MGU, Ser. 9: Filologiya* [Bulletin of the MSU, Ser. 9: Philology], 2000. 2, p.127—145.

-
- [6] **Hans-Georg Gadamer**. Truth and Method: Fundamentals of Philosophical Hermeneutics // Wahrheit und Methode. Grundzüge einer philosophischen Hermeneutik. Tübingen: J. C. B. Mohr (Paul Siebeck) 1960. 486 p.
- [7] **Gottlob Frege**. Sense and denotation. Semiotics and computer science. // Über Sinn und Bedeutung. In: Zeitschrift für Philosophie und philosophische Kritik. Band 100, 1892, P.25 – 50.
- [8] **Carnap R., Bar-Hillel Y.** An outline of theory of semantic information // Techn. Report of Res. Lab. Electr. – 1952. No. 247, 49 p.
- [9] **Wittgenstein L.** Philosophical Studies // Philosophische Untersuchungen / Philosophical Investigations. Translated by G.E.M. Anscombe. Oxford: Basil Blackwell, 1953, 232 p.
- [10] **Baclawski, K., Bennett, M., Berg-Cross, G., Fritzsche, D., Schneider, T., Sharma, R., ... Westerinen, A.** (2018). Ontology Summit 2017 Communique: AI, Learning, Reasoning and Ontologies. *Applied Ontology*, 13(1), 3–18. DOI: 10.3233/AO-180200.
- [11] **Fritzsche, D., Gruninger, M., Baclawski, K., Bennett, M., Berg-Cross, G., Obrst, L., ... Westerinen, A.** (2017). Ontology Summit 2016 Communique: Framing the Conversation: Ontologies within Semantic Interoperability Ecosystems. *Applied Ontology*, 12(2), 91–111. DOI: 10.3233/AO-170181.
- [12] **Gruninger, M., Obrst, L., Baclawski, K., Bennett, M., Brickley, D., Berg-Cross, G., ... Yim, P.** (2014). Ontology Summit 2014 Communique: The Semantic Web and Big Data meet Applied Ontology. *Applied Ontology*, 9(2), 155-170. DOI: 10.3233/AO-140135.
- [13] **Kamnev A.** The Application Programming Interface of the Geometric Kernel C3D, its application and the main difference from the API of the Kompas-3D System [In Russian]. CAD and Graphics [SAPR i grafika]. 2016. 5:36-38. <https://sapr.ru/article/25210>.
- [14] Module Computer numerical control. Turning. [Modul' ChPU. Tokarnaja obrabotka] [In Russian]: <https://kompas.ru/kompas-3d/application/machinery/module-chpu>.
- [15] Module Computer numerical control. Milling. [Modul' ChPU. Frezernaja obrabotka] [In Russian]: <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo>.
- [16] **Shchekin AV.** Design and technological parametrization as a part of an integrated CAM System [In Russian]. Information Technologies [Informatsionnyie tehnologii]. 2019; 25(7): 34-54.
- [17] **Evgenev GB.** Industry 5.0 as integration of the Internet of Knowledge and the Internet of Things [In Russian]. *Ontology of designing*. 2019; 9(1): 7-23. DOI: 10.18287/2223-9537-2019-9-1-7-23.
- [18] **Andrichenko AN.** Tendencies and the status in the field of reference data management in the engineering industry [In Russian]. *Ontology of designing*. 2012; 2(4): 25-35.
- [19] **Evgenev GB.** Multi-agent systems of semi-automatic design based on object-functions model in engineering [In Russian]. *Ontology of designing*. 2020; 10(1): 50-62. DOI: 10.18287/2223-9537-2020-10-1-50-62.
-

About the author

Alexander V. Shchekin (b. 1976) graduated from the Mordovia State University (Saransk-city) in 1998. Chief of the Automation of CNC Programming Research Laboratory, National Research Mordovia State University. Research interests: CAD, CNC-programming automation, CAM (Computer-Aided Manufacturing), ontological engineering. ResearcherID: F-4689-2019, ORCID: <https://orcid.org/0000-0001-5209-166X>, schekin@inbox.ru.

Received May 05, 2020. Revised May 25, 2020. Accepted June 08, 2020.
