Технология и технологические машины

УДК 004.624

Онтологическая методология создания интеллектуальных систем в машиностроении

Г.Б. Евгенев

МГТУ им. Н.Э. Баумана, 105005, Москва, Российская Федерация, 2-я Бауманская ул., д. 5, стр. 1.

An ontological methodology for developing intelligent systems in mechanical engineering

G.B. Evgenev

Bauman Moscow State Technical University, building 1, 2-nd Baumanskaya str., 5, 105005, Moscow, Russian Federation.

e-mail: g.evgenev@mail.ru

В настоящее время для решения проблемы повышения эффективности и конкурентоспособности машиностроительного производства необходима разработка новой методологии создания информационных систем поддержки компьютерно-интегрированных производств на соответствующих этапах жизненного цикла изделий (автоматизированного конструирования и проектирования технологических процессов). Основой для построения таких систем могут стать онтологии, которые до сих пор в области машиностроения применялись для представления и обработки данных. В статье предлагается совместно использовать предметные онтологии и онтологии задач для создания многоагентных систем проектирования и управления в машиностроении. Цель такого подхода состоит в получении синергетического эффекта от интеграции онтологических и многоагентных методов, что позволяет сократить трудоемкость создания и использования сложных интеллектуальных систем. Доказано, что применение этой методологии позволяет на прядок сократить трудоемкость создания конструкторско-технологических систем интеллектуального проектирования, доведя их эксплуатацию до полуавтоматического уровня.

Ключевые слова: онтология, многоагентные системы, автоматизация конструирования, автоматизация проектирования технологических процессов, компьютерно-интегрированное производство.

To increase the efficiency and competitiveness of engineering production, it is necessary to develop a new methodology for creating information systems supporting computer-integrated manufacturing at corresponding stages of the product life cycle (computer-aided engineering and design processes). Such systems can be constructed on the basis of ontologies that have been used in mechanical engineering for data representation and processing. In this paper, object and task ontologies are used jointly for developing multi-agent design and management systems to be implemented in engineering. The proposed approach provides a synergetic effect due to integrating the ontology and multi-agent methods, which makes it possible to improve the efficiency of developing and using complex intelligent systems. The results of study proved that the application of this methodology can significantly reduce the complexity of intelligent design systems so that they could be used semi-automatically.

Keywords: ontology, multi-agent systems, design automation, computer-aided design, manufacturing process, computer-integrated manufacturing.

Очень важным для экономики нашей страны является рост производительности труда в машиностроении, в том числе на этапах конструкторско-технологического проектирования и планирования для выпуска современной конкурентоспособной продукции. При этом необходимо создание и освоение новых наукоемких базовых информационных технологий.

В качестве базовой концепции здесь целесообразно использовать принцип «трех И»: ***И**нтеграция*, ***И**нтеллектуализация*, ***И**ндивидуализация*.

Интеграция должна обеспечить сокращение длительности цикла проектирование-производство за счет минимизации потерь времени при переходе от одной фазы жизненного цикла изделия к другой. В настоящее время в данном направлении достигнуты наибольшие успехи. Однако при этом используются простейшие методы перекодировки обменных файлов в смежных системах, разработанных на основе различных методов.

Интеллектуализация призвана сократить трудоемкость процесса проектирования за счет автоматизации принятия решений с использованием баз знаний. Одновременно может быть повышено качество проектирования благодаря использованию наиболее передовых знаний. Этот принцип особенно эффективен в области типового вариантного проектирования изделий, например в области электромашиностроения. К сожалению, в настоящее время данный принцип практически не реализован.

Индивидуализация тесно связана с интеллектуализацией, так как должна позволить специалистам, не владеющим информационными технологиями, изменять и дополнять базы данных и знаний применительно к специфике своего производства.

До настоящего времени в области машиностроения онтологии использовались лишь для представления и обработки данных [1–6]. Создание методологии, интегрирующей онтологический и многоагентный подходы, позволит получить синергетический эффект сокращения трудоемкости создания и использования сложных интеллектуальных систем.

Онтология. Термин «онтология» происходит от древнегреческих слов *онтос* — сущее и *логос* — учение. С точки зрения проблем, связанных с искусственным интеллектом (ИИ), *онтология* — это эксплицитная (явная) спецификация концептуализации знаний [7].

В связи с необходимостью явной спецификации процессов функционирования онтологии принято рассматривать онтологические системы. Под *формальной моделью онтологической системы* Σ° понимают [7] триплет вида

$$\Sigma^\circ = <O^{\text{meta}}, \{O^{d\&f}\}, \Xi^{\text{inf}}>,$$

где O^{meta} — онтология верхнего уровня (метаонтология); $\{O^{d\&f}\}=\{O^d\}\cup\{O^f\}$ — множество предметных онтологий O^d и онтологий задач O^f предметной области; Ξ^{inf} — модель машины вывода, ассоциированная с онтологической системой Σ°.

В модели Σ° имеются три онтологические компоненты:

- метаонтология;
- предметная онтология;
- онтология задач.

Метаонтология оперирует общими концептами и отношениями, которые не зависят от конкретной предметной области. Метаонтология должна содержать концепты и отношения, необходимые как для предметной онтологии, так и для онтологии задач. Последние в совокупности должны обеспечивать построение операциональной модели M предметной области [7]. На основе этой модели выполняется преобразование исходных данных *In*, необходимых для автоматизированного проектирования изделий и технологических процессов их изготовления, в выходные данные *Out*, содержащие модель результатов инженерного труда.

Предметная онтология O^d содержит понятия, описывающие конкретную предметную область и отношения между ними.

Онтология задач O^f содержит функции, с помощью которых выполняется преобразование входных данных *In* операциональной модели M в выходные данные *Out*.

Каждое понятие связывается с определенным методом, представляющим собой подсистему онтологии задач. Такая пара носит название *агент*. Машина вывода Ξ^{inf} онтологической системы инженерных знаний опирается на сетевое представление агентов, образующих метасистему. Функционирование ее связано с двумя процессами — структурным и параметрическим синтезом.

Многоагентные системы. Важнейшей методической основой для реализации упомянутых концепций является теория многоагентных систем (МАС).

При построении искусственных агентов минимальный набор базовых характеристик должен включать следующие [8, 9]: *активность, реактивность, автономность, общительность и целенаправленность*.

Рассмотрим использование МАС применительно к интеллектуальным системам автоматизации технологических процессов и производств. Здесь в качестве агентов выступают модели сборочных единиц, деталей и их элементов, а также технологических процессов. Активность агентов заключается в необходимости решения двух категорий задач — структурного и параметрического синтеза. Структурный синтез заключается в выборе структуры подчиненных объектов, а параметрический — в генерации значений собственных свойств, в результате чего из класса объектов, представленных в форме агента, генерируется один экземпляр, который и включается в проект.

Реактивность агентов обеспечивает решение упомянутых задач за счет обмена информацией между агентами непосредственно или через базу данных.

Автономность агентов основывается на встроенных в них методах, в которых содержатся инженерные знания по различного рода расчетам, а также геометрические и графические знания в форме параметризованных моделей, обеспечивающих генерацию трехмерных образов и чертежей.

Общительность агентов имеет вертикальную

и горизонтальную составляющие. Вертикальная составляющая включает обмен данными по иерархии «целое — часть» и «род — вид», а горизонтальная — обмен между конструктивно сопряженными, но не подчиненными друг другу по иерархии агентами.

Целенаправленность агентов определяется необходимостью реализации проекта, удовлетворяющего техническим требованиям заказчика, а также требованиям, накладываемым разработчиком.

Обобщенная модель класса искусственных агентов приведена на рис. 1. Любой агент представляет собой открытую систему, помещенную в некоторую среду [8]. В данном случае этой средой является проект, формируемый в базах данных, в качестве которых целесообразно использовать базу данных объектного типа для представления модели изделия (*внутренняя среда*), и реляционную базу данных для поиска стандартных и покупных изделий, свойств материалов и т. п. информации (*внешняя среда*). Внешняя среда, как правило, является сетевой.

Свойства агента могут принадлежать трем различным категориям: импортируемым, экспортируемым и внутренним. Импортируемые свойства являются *рецепторами* агента, формирующими его систему *восприятия*; экспортируемые свойства агента — его *эффекторами*, функция которых состоит в воздействии на среду, т. е. на состояние проекта.

Свойства агента всех трех категорий образуют его *память*, в которой хранится текущее состояние агента (см. рис. 1).

Процессор агента (см. рис. 1) формируют его методы, обеспечивающие объединение и переработку разнородных данных, выработку соответствующих реакций на информацию о состоянии среды (проекта), принятие решений о выполнении тех или иных действий [8]. В целом процессор определяет *поведение* агента, которое можно наблюдать, используя инспектор модели агента. За состоянием свойств агента инженер следит с помощью инспектора либо в графическом окне, в котором отображаются сгенерированные чертежи и другая геометрическая информация.

Многоагентная система как и любая другая направленная система может быть представлена следующей шестеркой:

$$\text{MAC} = \{Ind, Prp, Atr, Inp, Out, Str\}.$$

Здесь *Ind* — наименование системы; *Prp* — цели системы; *Atr* — общесистемные характеристики; *Inp* — вход системы; *Out* — выход системы; *Str* — структура системы, $Str = \{E, R\}$, где E — компонент системы, R — связи компонентов.

Для структурного моделирования МАС могут использоваться диаграммы классов UML. Унифицированный язык моделирования (Unified Modeling Language — UML) является графическим языком для визуализации, специфицирования, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению. UML можно использовать в качестве компоненты метаонтологии O^{meta}.

Диаграммы классов UML соответствуют статическому виду системы с точки зрения проектирования. Если такие диаграммы включают активные классы, то они соответствуют статическому виду системы при ее проектировании [8]. Однако модели классов UML не в полной мере удовлетворяют потребностям моделирования МАС. Классическая модель агента приведена на рис. 1, *а* [8]. На рис. 1, *б* на примере привода представлена модифицированная модель класса объектов в нотации UML. На рисунке видно, что процессор агента соответствует операциям класса, а память агента — значениям атрибутов класса. Несоответствие заключается в том, что агент имеет импортируемые и экспортируемые свойства, а в нотации класса UML это не предусмотрено. Знаком «+» в UML отмечаются свойства, видимые извне. Простейшим решением этой проблемы может быть пометка импортируемых свойств знаком «>», а экспортируемых — знаком «<»,

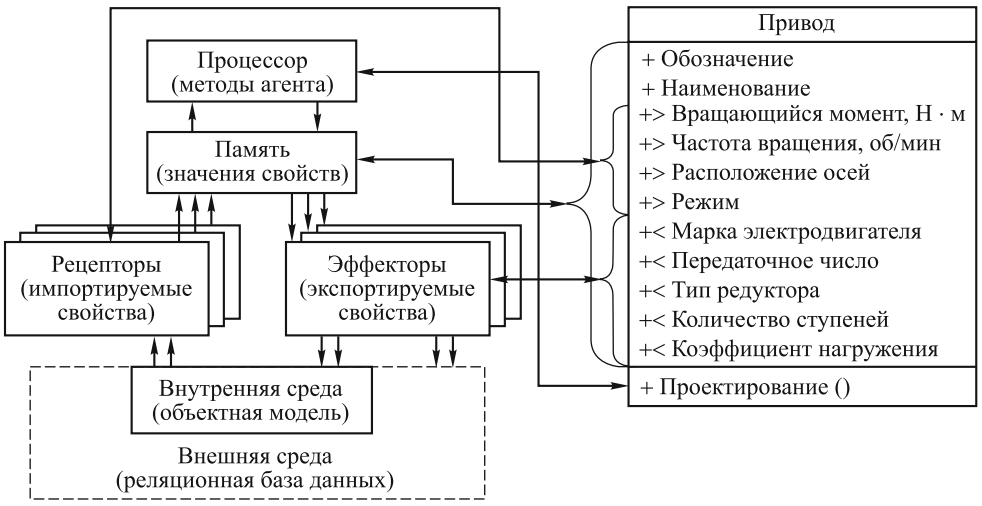

Рис. 1. Архитектура агента:
а — классическая модель; *б* — модифицированная модель

как это показано на рис. 1, *б*. При этом агент принимает форму *объект-функции*.

Диаграмма классов объект-функций представляет, с одной стороны, предметную онтологию O^d, а с другой, определяет модель машины вывода Ξ^{inf}.

Онтология задач. При использовании UML компонентами онтологии задач являются методы объектов. Однако используемый в UML для построения методов традиционный алгоритмический подход не соответствует явной спецификации концептуализации знаний с точки зрения непрограммирующего пользователя. В связи с этим онтологию задач следует строить с использованием баз знаний.

Для построения онтологий задач была разработана технология экспертного программирования [8]. Очевидно, что простейшей системой программирования была бы такая, в которой использовалась бы команды одного типа с несколькими разновидностями. С точки зрения объектно-ориентированного подхода это означает, что необходимо найти такой «суперкласс» объектов, который был бы способен охватить все проблемы программирования.

Разработка любой сложной, в том числе программной, системы должна начинаться с функционального анализа и моделирования системы в целом и всех ее подсистем вплоть до неделимых элементов. Для этой цели разработана методология IDEF0, представляющая собой совокупность методов, правил и процедур, предназначенных для построения функциональной структуры сложных иерархических систем. Методологию IDEF0 целесообразно включить в состав метаонтологии O^{meta}.

Основной принцип, заложенный в функциональное моделирование систем, состоит в их пошаговой нисходящей декомпозиции до уровня, необходимого для целей моделирования. При этом на всех уровнях используются функциональные блоки, принадлежащие к одному и тому же классу, который можно назвать объект-функция.

В экспертном программировании в качестве суперкласса используется объект-функция IDEF0. Графически такая объект-функция представляется в форме прямоугольника. Каждая из четырех сторон прямоугольника имеет определенное назначение: левая — входы, правая — выходы, верхняя — управление, нижняя — механизмы. Входы представляют собой информацию, необходимую для выполнения функции, и в результате ее выполнения преобразуются в выходы. Входы показывают все характеристики, которые необходимы для выполнения функции и она не может быть выполнена без получения их значений. Управление описывает условие, оказывающее влияние на выполнение функции, но само не подвергается переработке.

К нижней части изображения объект-функции присоединяются стрелки механизмов, обозначающие программное средство, обеспечивающее выполнение функции. Входы и выходы показывают, что делается функцией, управление — почему это делается, а механизмы — с помощью чего делается.

Если обозначить через $\boldsymbol{X} = (x_1, x_2, \ldots, x_m)$ вектор входных переменных, через $\boldsymbol{Y} = (y_1, y_2, \ldots, y_n)$ вектор выходных переменных, а через $\boldsymbol{F}$ вектор-функцию, реализуемую механизмом, то получим выражение объект-функции, эквивалентное традиционному математическому:

$$\boldsymbol{Y} = \boldsymbol{F}(\boldsymbol{X}).$$

Однако в отличие от математических функций, допускающих использование в качестве переменных только числовые величины, в объект-функциях могут использоваться как числовые, так и нечисловые переменные.

При построении диаграмм в IDEF0 функциональные блоки соединяются с помощью стрелок, идущих от выхода одного блока к входу и (или) управлению другого. Такая диаграмма с точки зрения искусственного интеллекта представляет собой семантическую сеть, т. е. граф с помеченными с помощью идентификаторов или наименований вершинами (объект-функциями) и ребрами. С математической точки зрения диаграмма эквивалентна сложной функции

$$\boldsymbol{Y} = ((\boldsymbol{F}_1(\boldsymbol{X}_1), \boldsymbol{F}_2(\boldsymbol{X}_2), ..., \boldsymbol{F}k(\boldsymbol{X}k)).$$

В интеллектуальных системах проектирования подобные блоки или модули действий (с точки зрения UML), или функциональные блоки (с точки зрения IDEF0), или правила-продукции (с точки зрения искусственного интеллекта) принято называть модулями знаний (МЗ).

Неструктурированная совокупность МЗ в определенной прикладной области представляет собой базу знаний этой области, аналогичную базе знаний продукционной системы.

Наименования и имена входных, управляющих и выходных переменных МЗ должны выбираться из словаря базы знаний, представленного ниже.

Словарь в совокупности со значениями содержащихся в нем переменных выполняет в экспертном программировании функцию рабочей памяти продукционной системы.

Механизмы МЗ должны обеспечивать реализацию всех функций, которые могут потребоваться при формировании баз знаний. В число таких функций входят следующие основные:

- вычисление по формулам (в том числе присвоение значений переменным),
- определение значений по таблицам,
- выбор значений из баз данных,
- обновление значений в базах данных,
- занесение значений в базы данных,
- вычисление значений с использованием подпрограмм,
- вычисление значений с помощью методов, сгенерированных из модулей знаний,
- вычисление значений с помощью исполняемых exe-модулей или dll-библиотек, сгенерированных другими системами.

Внешнее представление модуля (формулы) приведено на рис. 2. Оно автоматически генерируется инструментальной средой системы ЭксПро.

С точки зрения структуры IDEF0 здесь управляющей переменной является «Вид СЕ», а входными

Словарь базы знаний

Имя	Наименование	Тип
Mt	Момент крутящий на выходном валу, Н×м	REAL
nt	Частота вращения на выходе, об/мин	REAL
RaspOs$	Расположение вход. и выход. осей	STRING
Nct	Количество ступеней	INTEGER
TypRed$	Тип редуктора	STRING
MarEd$	Марка электродвигателя	STRING
nh	Частота вращения на входе, об/мин	REAL
urz	Передат. отношение редуктора зад.	REAL
Nz	Мощность электродвигателя потребная, кВт	REAL
Cena	Цена электродвигателя, руб.	REAL

переменными — «Частота вращения на входе, об/мин» и «Частота вращения на выходе, об/мин». Значения этих переменных используются при расчете выходной переменной «Передаточное число», которая равна отношению входных переменных.

Если рассматривать этот МЗ как правило-продукцию, то он эквивалентен следующему предложению: «если вид сборочной единицы *Редуктор* и частота вращения на выходе, об/мин (nt), больше 0, то передаточное число вычисляется по формуле Urz = nh/nt».

Возможно с помощью одного МЗ присваивать значения переменным и производить вычисления по набору взаимосвязанных формул. При этом предшествующие выходные переменные могут использоваться для определения последующих выходных переменных.

Для построения онтологий задач МЗ должны объединяться в методы. На рис. 1 метод агента назван процессором. Генерация методов производится автоматически с помощью инструментальной среды ЭксПро [8].

Система конструирования. В настоящее время системы автоматизированного конструирования (CAD системы) не соответствуют требованиям онтологической методологии. В худшем случае с помощью CAD системы генерируется геометрическая 3D модель, которая передается в систему программирования обработки на станках с ЧПУ (CAM систему) чаще всего с помощью метафайла в стандарте IGES. С помощью 3D модели формируются чертежи изделий со всеми необходимыми техническими требованиями, но эта информация недоступна для дальнейшей компьютерной обработки.

МЗ: «PR3» – Расчет числа передаточного
Предусловия запуска

Имя	Наименование	Тип	Условие
VidSE nt	Вид СЕ Частота вращения на выходе, об/мин	STRING REAL	Редуктор (0,)

Входные свойства

Имя	Наименование	Тип	Значение
nh	Частота вращения на входе, об./мин	REAL	
nt	Частота вращения на выходе, об./мин	REAL	

Механизм – Формула
Urz = nh/nt

Выходные свойства

Имя	Наименование	Тип	Значение
Urz	Передаточное число	REAL	

Рис. 2. Символьное представление объект-функции

В лучшем случае с помощью CAD системы генерируется полная объектная модель в стандарте STEP. Этот стандарт позволяет формировать полномасштабную предметную онтологическую модель изделий. Однако он сложен и требует высокой квалификации исполнителей. На практике стандарт STEP используется крупными предприятиями, для изделий которых разработаны базовые объектные модели.

Следует подчеркнуть, что стандарт STEP не связан с построением онтологии задач, использование которой превращает систему проектирования в интеллектуальную.

Для онтологической методологии необходима совместная разработка предметной онтологии и онтологии задач. Поскольку онтология должна быть основана на явной спецификации знаний, то эта спецификация должна использовать формы представления максимально приближенные к естественному языку.

Первым шагом методологии является построение многоагентной структуры в форме, максимально приближенной к диаграмме классов UML. Фрагмент такой диаграммы применительно к конструкторско-технологической онтологии для асинхронных электродвигателей представлен на рис. 3.

Корневым агентом конструкторской части диаграммы является «Изделие», а технологической части — «Технологический процесс», который зависит от конструкции изделия, что отражено пунктирной стрелкой. Изделие на диаграмме имеет две разновидности: «Сборочная единица» и «Деталь». Детали входят как части в сборочную единицу. Электродвигатель является одной из разновидностей сборочных единиц. В состав электродвигателя входят основные сборочные единицы «Ротор» и «Статор». Главной деталью статора является «Станина».

На диаграмме представлены основные разновидности технологических процессов: «Формообразование», «Обработка давлением», «Обработка резанием», «Сборка», «Получение покрытий» и прочие методы.

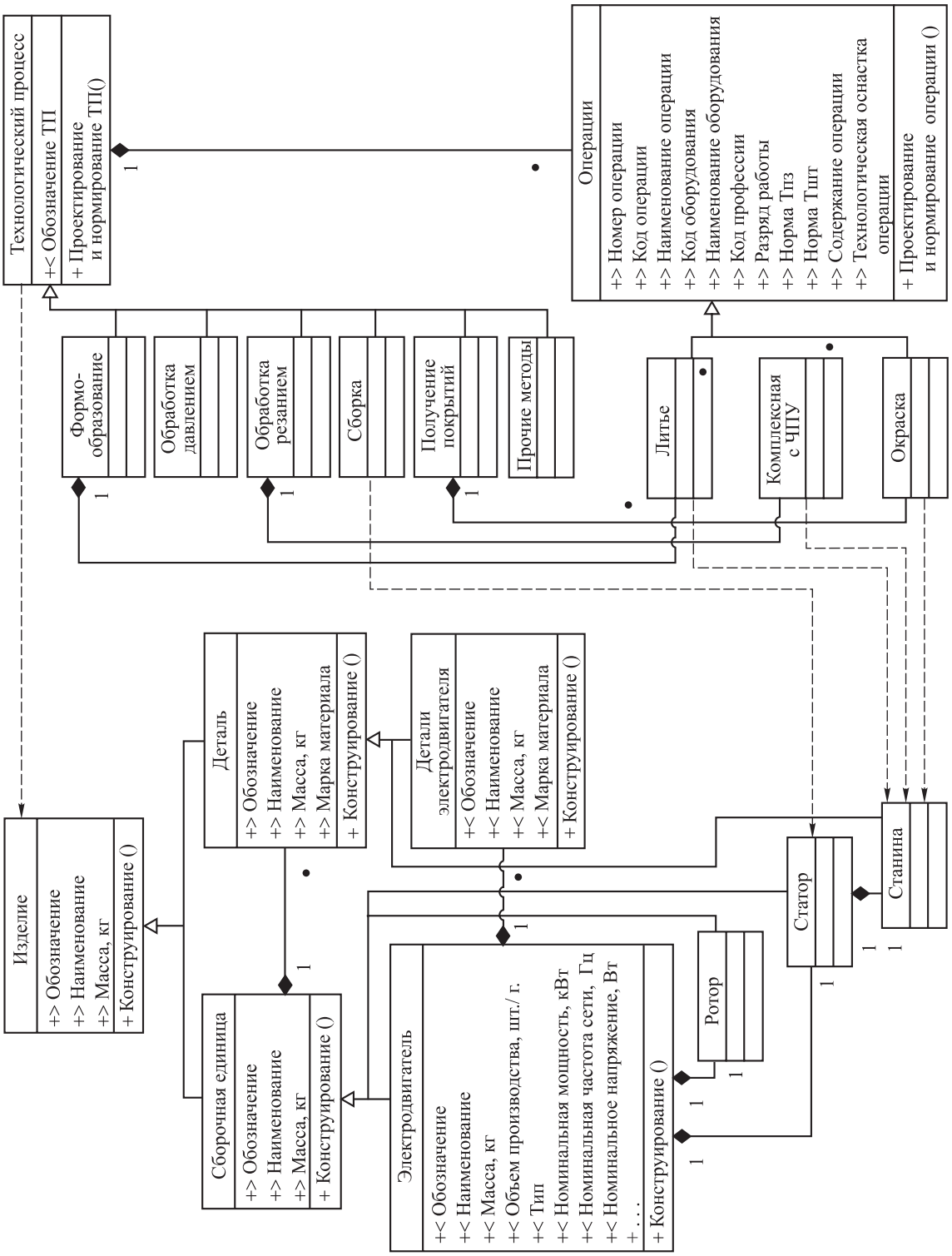

Рис. 3. Фрагмент конструкторско-технологической многоагентной онтологии для электродвигателя

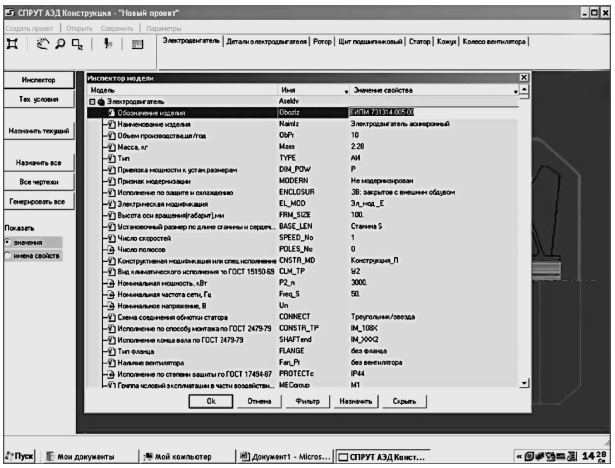

Рис. 4. Атрибуты электродвигателя и их значения
(Полноцветную версию см. http://www.izvuzmash.bmstu.ru)

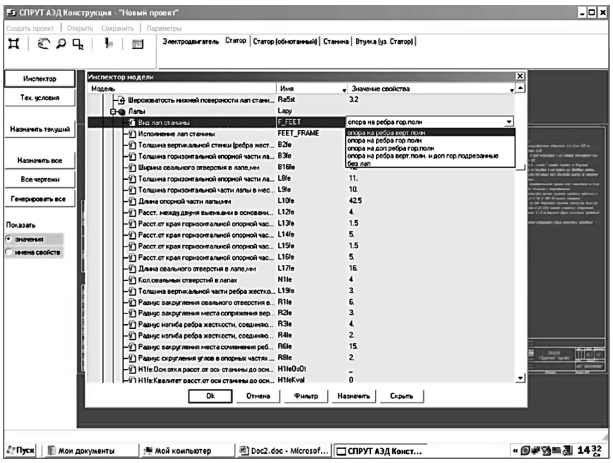

Рис. 5. Атрибуты компонента «Лапы» и их значения
(Полноцветную версию см. http://www.izvuzmash.bmstu.ru)

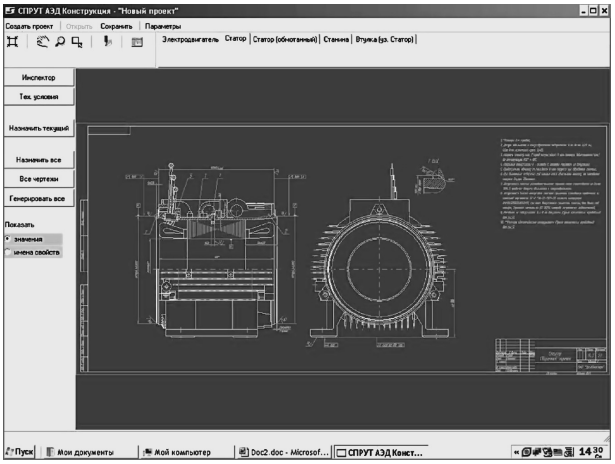

Рис. 6. Чертеж статора с лапами
(Полноцветную версию см. http://www.izvuzmash.bmstu.ru)

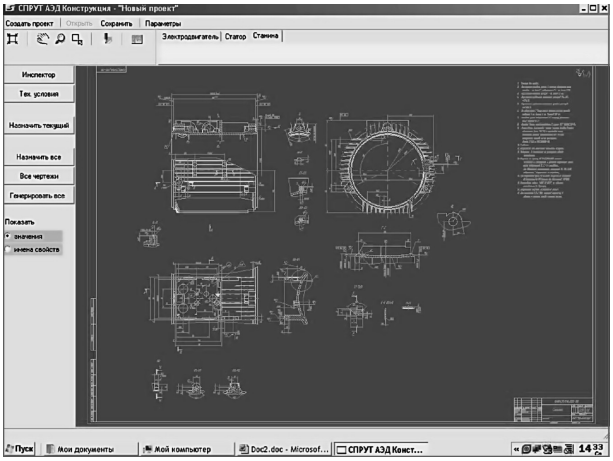

Рис. 7. Чертеж станины без лап
(Полноцветную версию см. http://www.izvuzmash.bmstu.ru)

Технологический процесс складывается из операций. На диаграмме приведены три разновидности операций: «Литье», которое является частью операций формообразования, «Комплексная с ЧПУ», входящая в состав операций обработки резанием, и «Окраска» из числа процессов получения покрытий. Атрибуты этих операций зависят от конструктивных характеристик станины. Для производства статора используется операция сборки.

Приведенная на рис. 3 диаграмма представляет собой «чертеж» онтологической МАС. Для ввода этого «чертежа» в компьютер используется инструментальная среда Sprut-X [8].

Типовой интерфейс прикладной интеллектуальной системы полуавтоматического конструирования, создаваемой с помощью Sprut-X, приведен на рис. 4–7. Система является полуавтоматической поскольку после ввода данных запрашиваемых компьютером она производит генерацию необходимого комплекта чертежей изделия.

В верхней части окна интерфейса автоматически формируется набор закладок, соответствующий уровням вхождения агентов в диаграмму классов (см. рис. 3). Набор закладок перестраивается в зависимости от выбора базового агента. На рис. 4 в качестве базового агента выбран «Электродвигатель». Справа от него представлены сборочные единицы и детали первого уровня вхождения. На рис. 5 базовым агентом является «Статор» с соответствующей сменой входящих агентов.

На левом поле интерфейса располагаются кнопки, соответствующие методам выбранного базового агента. Стандартным методом является «Инспектор», с помощью которого в правом поле отображается содержание предметной онтологической модели входящих агентов с указанием текущих значений атрибутов. С помощью иконок отмечается тип каждого атрибута: *V* — экспортируемый или внутренний, → — импортируемый тип.

В правом поле интерфейса после нажатия кнопки соответствующего метода производится автоматическая генерация чертежей сборочных единиц и деталей.

При этом при изменении значений экспортируемых атрибутов производится полная перегенерация всех взаимосвязанных чертежей. Так, после выбора

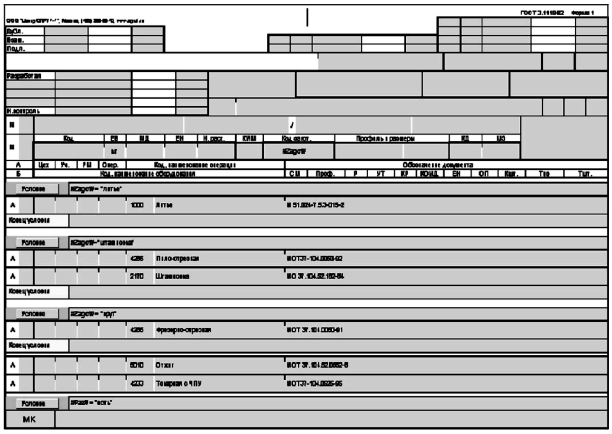

Рис. 8. Разработка онтологии задач структурного синтеза технологического процесса

значения атрибута «Вид лап станины» агента «Лапы» выполняется перегенерация чертежей сборочной единицы «Статор» и детали «Станина». Чертеж статора с лапами приведен на рис. 6, а на рис. 7 — станины без лап.

Система технологического проектирования. Онтологический подход был использован при создании системы автоматизации проектирования технологических процессов «Спрут-ТП» [8]. Для этого в качестве интерфейса в системе были применены стандартные по ЕСТД активные бланки маршрутных и операционных карт. Бланки содержат полный набор атрибутов агентов «Технологический процесс» и «Операция» из диаграммы классов, изображенных на рис. 3.

При расположении указателя в определенном поле бланка автоматически подключается метод информационной поддержки проектирования, позволяющий выбирать из базы данных ресурсов допустимые значения атрибута. При этом набор допустимых значений зависит от предварительно принятых решений. Так осуществляется параметрический синтез в процессе проектирования.

Для структурного синтеза необходимо создание базы знаний. Чтобы обеспечить возможность создания базы знаний на языке, доступном непрограммирующему пользователю, был использован модифицированный интерфейс маршрутной карты, представленный на рис. 8. В таких картах технологический процесс описывается с использованием стандартных строк типа А и типа Б. Строки типа А используются для задания параметров операций, а строки типа Б — для описания используемого оборудования. Чтобы получить возможность формирования диаграмм процессов в стандарте IDEF3, который целесообразно включить в метаонтологию, к числу стандартных технологических строк типа А и типа Б необходимо добавить строки, позволяющие задавать условия вхождения операций в итоговый технологический процесс. Эти условия должны позволять описывать логические связки типа исключающего ИЛИ.

Для задания логических связок в форму маршрутной карты введены строки типа «Условие» и «Конец условия». Эти строки в совокупности с заключенными между ними стандартными технологическими строками представляют собой аналог правила-продукции. Весь массив такой информации может рассматриваться как своеобразный аналог базы знаний продукционного типа с упорядочением правил во времени.

Форма ввода фрагмента диаграммы процесса обработки цилиндрических зубчатых колес приведена на рис. 8. В зависимости от значения переменной #Zagot# выбираются те или иные заготовительные операции, которые заносятся в итоговый документ. При #Zagot# = «литье» в технологический процесс включается операция «Литье». Если #Zagot# = «штамповка», то в состав процесса включаются операции «Пило-отрезная» и «Штамповка», а в случае #Zagot# = = «круг» включается операция «фрезерно-отрезная». Далее идут безусловно назначаемые операции «Отжиг» и «Токарная с ЧПУ».

Выводы

1. Разработана онтологическая многоагентная методология создания интегрированных интеллектуальных систем в машиностроении. Методология посредством интеграции на концептуальном уровне обеспечивает сокращение длительности цикла проектирование-производство за счет минимизации потерь времени при переходе от одной фазы жизненного цикла изделия к другой. Интеллектуализация позволяет сократить трудоемкость проектирования и повысить качество за счет автоматизации принятия решений.

2. В методологии при формировании баз данных и знаний используется ограниченный естественный язык, что позволяет привлекать к этому процессу непрограммирующих прикладных пользователей и значительно сократить трудоемкость процессов разработки и индивидуализации систем.

Литература

[1] Норенков И.П. Интеллектуальные технологии на базе онтологий. *Информационные технологии*, 2010, № 1, с. 17–23.

[2] Garcia L.E.R. Ontological CAD Data Interoperability Framework. *Proceedings of SEMAPRO 2010: The Fourth International Conference on Advances in Semantic Processing*, 2010, pp. 79–82.

[3] Odd A., Vasilakis G. *Building an Ontology of CAD Model Information. Geometric Modeling, Numerical Simulation, and Optimization*. Norway, SINTEF, 2007, pp. 11–41.

[4] Grüniner M., Deval A. A First-Order Cutting Process Ontology for Sheet Metal Parts. *In Proceeding of the 2009 Conference on Formal ontologies Meet industry. Frontiers in Artificial Intelligence and Applications*, Amsterdam, IOS Press, 2009, vol. 198, pp. 22–33.

[5] Patil L., Dutta D., Sriram R. Ontology-Based Exchange of Product Data Semantics. *IEEE Transactions on automation science and engineeting*, 2005, vol. 2, no. 3, pp. 213–224.
[6] Dartigues C., Ghodous P., Gruninger M., Pallez D., Sriram R. CAD/CAPP Integration using Feature Ontology. *Concurrent Engineering*, 2007, vol. 15, no. 2, pp. 237–249.
[7] Гаврилова Т.А., Хорошевский В.Ф. *Базы знаний интеллектуальных систем*. Санкт-Петербург, Питер, 2000. 384 с.
[8] Евгенев Г.Б. *Интеллектуальные системы проектирования*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2012. 420 с.
[9] Тарасов В.Б. *От многоагентных систем к интеллектуальным организациям: философия, психология, информатика*. Москва, Эдиториал УРСС, 2002. 352 с.

References

[1] Norenkov I.P. Intellektual'nye tekhnologii na baze ontologii [Intellectual Technologies on the Base of Ontologies]. *Informatsionnye tekhnologii* [Information Technology]. 2010, no. 1, pp. 17–23.
[2] Garcia L.E.R. Ontological CAD Data Interoperability Framework. *Proceedings of SEMAPRO 2010: The Fourth International Conference on Advances in Semantic Processing*, 2010, pp. 79–82.
[3] Odd A., Vasilakis G. *Building an Ontology of CAD Model Information. Geometric Modeling, Numerical Simulation, and Optimization*. Norway, SINTEF, 2007, pp. 11–41.
[4] Grüninger M., Deval A. A First-Order Cutting Process Ontology for Sheet Metal Parts. *In Proceeding of the 2009 Conference on Formal ontologies Meet industry. Frontiers in Artificial Intelligence and Applications*, Amsterdam, IOS Press, 2009, vol. 198, pp. 22–33.
[5] Patil L., Dutta D., Sriram R. Ontology-based exchange of product data semantics. *IEEE Transactions on Automation Science and Engineering*, 2005, vol. 2, no. 3, pp. 213–224.
[6] Dartigues C., Ghodous P., Gruninger M., Pallez D., Sriram R. CAD/CAPP integration using feature ontology. *Concurrent Engineering*, 2007, vol. 15, no. 2, pp. 237–249.
[7] Gavrilova T.A., Khoroshevskii V.F. *Bazy znanii intellektual'nykh system* [Knowledge Base Intelligent Systems]. Sankt-Peterburg, Piter publ., 2000. 384 p.
[8] Evgenev G.B. *Intellektual'nye sistemy proektirovaniia* [Intelligent systems design]. Moscow, Bauman Press, 2012. 420 p.
[9] Tarasov V.B. *Ot mnogoagentnykh sistem k intellektual'nym organizatsiiam: filosofiia, psikhologiia, informatika* [Of multi-agent systems for intelligent organizations: philosophy, psychology, computer science]. Moscow, Editorial URSS publ., 2002. 352 p.

Статья поступила в редакцию 17.12.2013

Информация об авторе

ЕВГЕНЕВ Георгий Борисович (Москва) — доктор технических наук, профессор кафедры «Компьютерные системы автоматизации производства». МГТУ им. Н.Э. Баумана (105005, Москва, Российская Федерация, 2-я Бауманская ул., д. 5, стр. 1, e-mail: g.evgenev@mail.ru).

Information about the author

EVGENEV Georgiy Borisovich (Moscow) — Dr. Sc. (Eng.), Professor of «Computer Systems of Automated Production» Department. Bauman Moscow State Technical University (BMSTU, building 1, 2-nd Baumanskaya str., 5, 105005, Moscow, Russian Federation, e-mail: g.evgenev@mail.ru).