

ЭКСПЕРТОПЕДИЯ КАК СРЕДСТВО СОЗДАНИЯ ОНТОЛОГИЧЕСКОГО ИНТЕРНЕТА ЗНАНИЙ

Г.Б. Евгеньев

Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия
g.evgenyev@mail.ru

Аннотация

Предложена методология создания семантических вики-систем для представления знаний с использованием технологий искусственного интеллекта. Методология основана на многоагентных методах создания баз знаний и пригодна для разработки систем проектирования и управления для цифровых интеллектуальных производств. При формировании внешнего представления баз знаний могут использоваться национальные языки. Проанализированы схемы ввода знаний в компьютер. На основе технологии экспертного программирования для создания баз знаний предложена семантическая вики-система, которая может быть названа «Экспертопедией». Возможность замены словаря базы знаний позволяет обеспечить выполнение концептуального требования вики-систем: использование любых языков. В базе знаний имеются возможности добавления модулей, выбора модуля-аналога, трансляции модуля знаний на один из языков программирования, тестирования полученного результата, а также определения входимости модуля в базы знаний и удаления выбранного модуля. Эти действия могут выполняться любым носителем знаний, что позволяет реализовать вторую концептуальную основу вики-систем: обеспечение возможности независимого пополнения и корректирования содержания системы. Разработана метаонтология Экспертопедии. Подробно рассмотрены методы создания многоагентных баз знаний для проектирования и управления в машиностроении. Приведена архитектура агента САПР. Даны примеры применения многоагентных систем для создания интеллектуальных систем полуавтоматического проектирования изделий машиностроения. Описана технология экспертного программирования.

Ключевые слова: экспертное программирование, цифровые производства, Интернет знаний, Интернет вещей, интеллектуальные системы.

Цитирование: Евгеньев, Г.Б. Экспертопедия как средство создания онтологического Интернета знаний / Г.Б. Евгеньев // Онтология проектирования. – 2019. – Т.9, №3(33). – С.307-320. – DOI: 10.18287/2223-9537-2019-9-3-307-320.

Введение

В настоящее время в мире происходит четвёртая промышленная революция (4ПР) [1, 2]. Суть новой революции заключается в том, что материальный мир соединяется с виртуальным, в результате чего рождаются новые социо-киберфизические системы, объединённые в одну цифровую экосистему. Фундаментом 4ПР является Интернет вещей (ИнВ). ИнВ — концепция вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключаящее из части действий и операций необходимость участия человека [3, 4]. 4ПР охватывает автоматизацию всех процессов, включая цифровое проектирование изделия, создание его виртуальной копии, совместную работу инженеров-конструкторов и технологов в едином цифровом пространстве, то есть,

когда изделия ещё не являются вещами, а существуют в виртуальном мире в виде информационных моделей.

Рассматриваются в совокупности два мира: мир виртуальный, реализуемый Интернетом знаний (ИнЗ), и мир реальный, реализуемый ИнВ [5]. ИнЗ целесообразно строить на онтологической методологии [6]. Метаонтология ИнЗ включает предметную онтологию, онтологию задач и онтологию оптимизации. Предметная онтология состоит из иерархии понятий, их определений и атрибутов, а также связанных с ними аксиом и правил вывода. Онтология задач включает задачи структурного и параметрического синтеза моделей изделий и процессов. С её помощью обеспечивается генерация цифровых моделей процессов и производства. Онтология оптимизации включает компоненты однокритериальной и многокритериальной оптимизации.

До цифровой революции носителями знаний были письменные источники. В результате цифровой революции носителями знаний стали программные средства, которые изначально строились на алгоритмической основе [6]. Непрограммирующийся носитель знаний не мог вводить их в компьютер. Необходимы были посредники: разработчик (алгоритмист) и программист (рисунок 1а). В начале развития искусственного интеллекта (ИИ) были созданы достаточно сложные языки описания знаний [7, 8]. В результате схема процесса ввода знаний в компьютер не изменилась [9] (рисунок 1б). Опять нужны были посредники в виде инженеров по знаниям и программистов. При этом между участниками процесса, как правило, имел место семантический разрыв. Каждый из посредников представлял знания, вводя свои соображения. В результате знания не в полной мере отражали знания их носителя. Цифровая революция должна кардинальным образом изменить эту схему и дать возможность непрограммирующему носителю знаний вводить их в компьютер без посредников (рисунок 1в). Это стало возможным, используя методологию *экспертного программирования* [10]. В этой методологии знания описываются на языке *деловой прозы*, максимально приближенном к литературному языку, но формализованном настолько, что имеется возможность автоматической генерации программных средств, соответствующих исходным текстам.

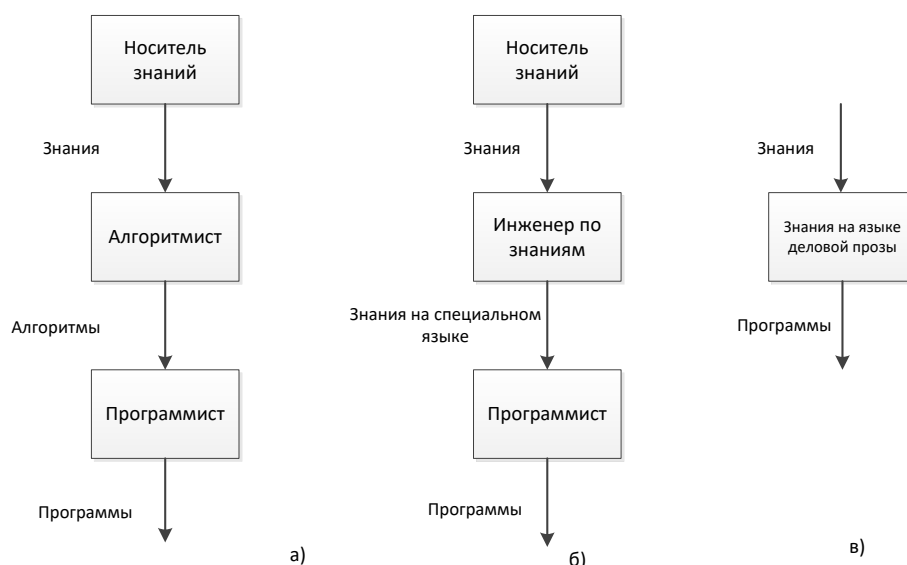


Рисунок 1 - Схемы ввода знаний: а) процедурное программирование, б) традиционные интеллектуальные системы, в) системы использования языка деловой прозы

1 Википедия и Экспертопедия

Наиболее распространённой компьютерной энциклопедией в настоящее время является «Википедия» — общедоступная многоязычная универсальная Интернет-энциклопедия со свободным контентом [11]. Концептуальными принципами её являются многоязычность и возможность пользователей пополнять и корректировать содержание. Создаваемые на этой основе традиционные вики-системы имеют ряд недостатков, к числу которых, в частности, относится недостаточная согласованность содержимого. В вики-системах из-за частого дублирования данных одна и та же информация может содержаться на нескольких разных страницах. При изменении этой информации на одной вики-странице пользователи должны следить за тем, чтобы данные были обновлены также и на всех остальных страницах.

Другой недостаток — сложность доступа к знаниям, имеющимся в вики-системах. Большие вики-сайты содержат тысячи страниц. Выполнение сложных поисковых запросов и сравнение информации, полученной с разных страниц в традиционных вики-системах, является трудоёмкой задачей. Традиционные вики используют плоские системы классификации (тэги), либо классификаторы, организованные в таксономию. Невозможность использования типизированных свойств порождает огромное количество тэгов или категорий.

В этой связи появились семантические вики — веб-приложения, использующие машинообрабатываемые данные со строго определённой семантикой [12]. Обычные вики-системы заполняются структурированным текстом и нетипизированными гиперссылками. Семантические вики-системы позволяют указывать тип ссылок между статьями, тип данных внутри статей, а также информацию о страницах (метаданные). Разновидность таких систем, основанных на технологии экспертного программирования для создания баз знаний (БЗ), может быть названа «Экспертопедией» (рисунок 2).

Семантика Экспертопедии определяется метаонтологией. Коренным метаобъектом в этом случае является БЗ, которая имеет свое наименование, имя-идентификатор и версию. Методы, подключенные к БЗ, позволяют производить сортировку и поиск модулей знаний (МЗ), из которых она состоит. Подмножество МЗ, объединённое в семантическую сеть, является модулем, который можно экспортировать в другие БЗ, а также импортировать и производить слияние в единую семантическую сеть. В вики-системе БЗ представляет собой страницу. При этом БЗ может рассматриваться как модуль в составе другой БЗ, которая в этом случае состоит из набора страниц [5]. Возможность замены словаря БЗ позволяет обеспечить выполнение концептуального требования вики-систем: использования любых языков мира.

Функциями модуля по преобразованию данных могут быть формулы, таблицы, работа с базами данных (БД), генерации 3D моделей объектов, математические модели.

Разновидностью математических знаний, необходимых для выполнения расчётов, являются модели непрерывных

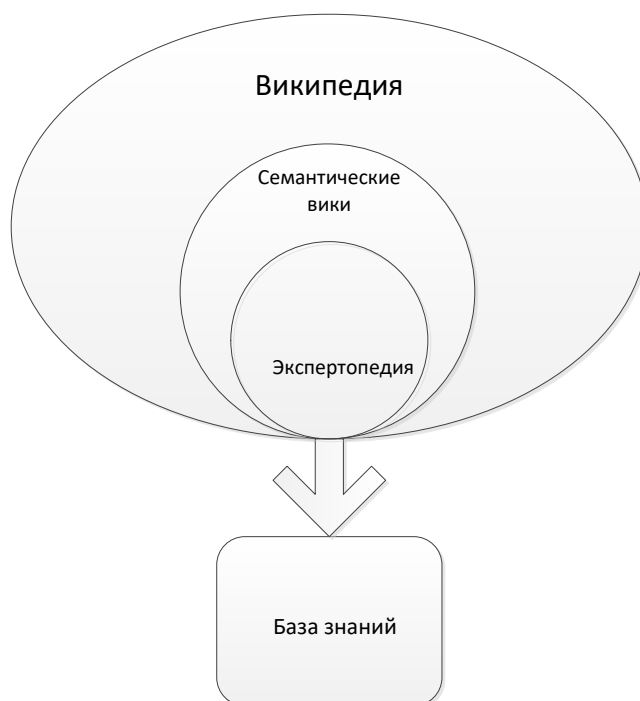


Рисунок 2 - Википедия и Экспертопедия

систем, основанные на дифференциально-алгебраических системах уравнений. Для реализации этих возможностей необходимо иметь средство, обеспечивающее: поддержку технологии объектно-ориентированного моделирования, совместимую с языком UML; удобное и адекватное описание модели на общепринятом математическом языке без написания какого-либо программного кода; автоматическое построение компьютерной модели, соответствующей заданной математической, с возможностью её автономного использования. В наибольшей мере этим требованиям отвечает пакет Model Vision Studium (MVS) [13]. Основным элементом MVS является активный динамический объект, который обеспечивает моделирование не только непрерывного поведения, но также дискретного или гибридного.

Семантические вики-системы позволяют указывать тип ссылок между статьями, тип данных внутри статей, а также информацию о страницах (метаданные). В Экспертопедии статьями являются БЗ, которые могут входить одна в другую в форме МЗ.

2 Многоагентные методы создания баз знаний проектирования

Многоагентные методы широко применяются в различных областях человеческих знаний [14, 15]. Основными концепциями дальнейшего развития систем автоматизации проектирования в машиностроении являются *интеграция, интеллектуализация и индивидуализация* [16]. Интеграция позволит ликвидировать перекодировку информации при переходе от одной фазы жизненного цикла изделия к другой. В итоге будут сформированы все данные, необходимые для систем планирования и управления производством. Интеллектуализация приведёт к сокращению трудоёмкости за счёт повышения уровня автоматизации, преобразования этих систем из пассивного инструмента в руках инженера в активного партнера, обеспечивающего автоматическое принятие решений и генерацию там, где это возможно, проектов изделий в целом или их узлов. Индивидуализация направлена на преобразование систем автоматизации из обезличенного программного продукта в персональное программное средство, наполненное индивидуальными знаниями экспертов без помощи программистов.

Многоагентная система (МАС) может быть представлена $MAC = \{Ind, Prp, Atr, Inp, Out, Str\}$. Здесь *Ind* — наименование системы; *Prp* — цели системы; *Atr* — общесистемные характеристики; *Inp* — вход системы; *Out* — выход системы; *Str* — структура системы. $Str = \{E, R\}$, где *E* — компоненты системы, а *R* — связи компонентов.

При использовании МАС для автоматизации проектирования (МАСАПР) наименование системы обычно содержит краткую характеристику её назначения, например «Система проектирования цилиндрических редукторов». Основные цели МАСАПР состоят в сокращении трудоёмкости, сроков и себестоимости проектирования, а также повышении его качества. Сокращение трудоёмкости, стоимости сопровождения и развития МАС осуществляется за счёт использования автоматической генерации программных средств, а также модульности, открытости и простоты модернизации систем проектирования.

Входная информация вводится в систему пользователем в начале проектирования, а выходная представляет собой проект, сформированный во внешней среде в результате работы МАС. Для структурного моделирования МАС могут быть использованы диаграммы классов UML. Классическая модель агента включает: процессор, в котором содержатся методы агента; память, содержащую значения свойств, которые делятся на рецепторы (импортируемые свойства) и эффекторы – свойства, экспортируемые во внутреннюю среду агента; внешние среды, представляющие реляционные БД [16]. Агент представляет собой *объект-функцию*, для представления модели которого может быть использована

модифицированная нотация класса UML. Процессор агента соответствует операциям класса, память агента – значениям атрибутов класса. Предложено пометить импортируемые (входные) свойства знаком >, а экспортируемые (выходные) – знаком <.

При создании интеллектуальных систем первой фазой проектирования является формирование номенклатуры и оргструктуры агентов. Номенклатуру агентов задают в словаре системы. Формирование оргструктуры агентов осуществляется в форме построения диаграммы классов UML. Применительно к созданию программных средств для машиностроения диаграммы классов эквивалентны конструкторским сборочным чертежам.

На рисунке 3 представлена диаграмма классов приводов. В эту диаграмму входит класс объектов «Редуктор», обладающий следующими входными свойствами: вращающий момент и частота вращения выходного вала редуктора. В число выходных свойств входят: вид редуктора, количество ступеней, расположение осей валов (которое для цилиндрических редукторов может принимать два значения: параллельное или соосное), время работы и вид нагружения.

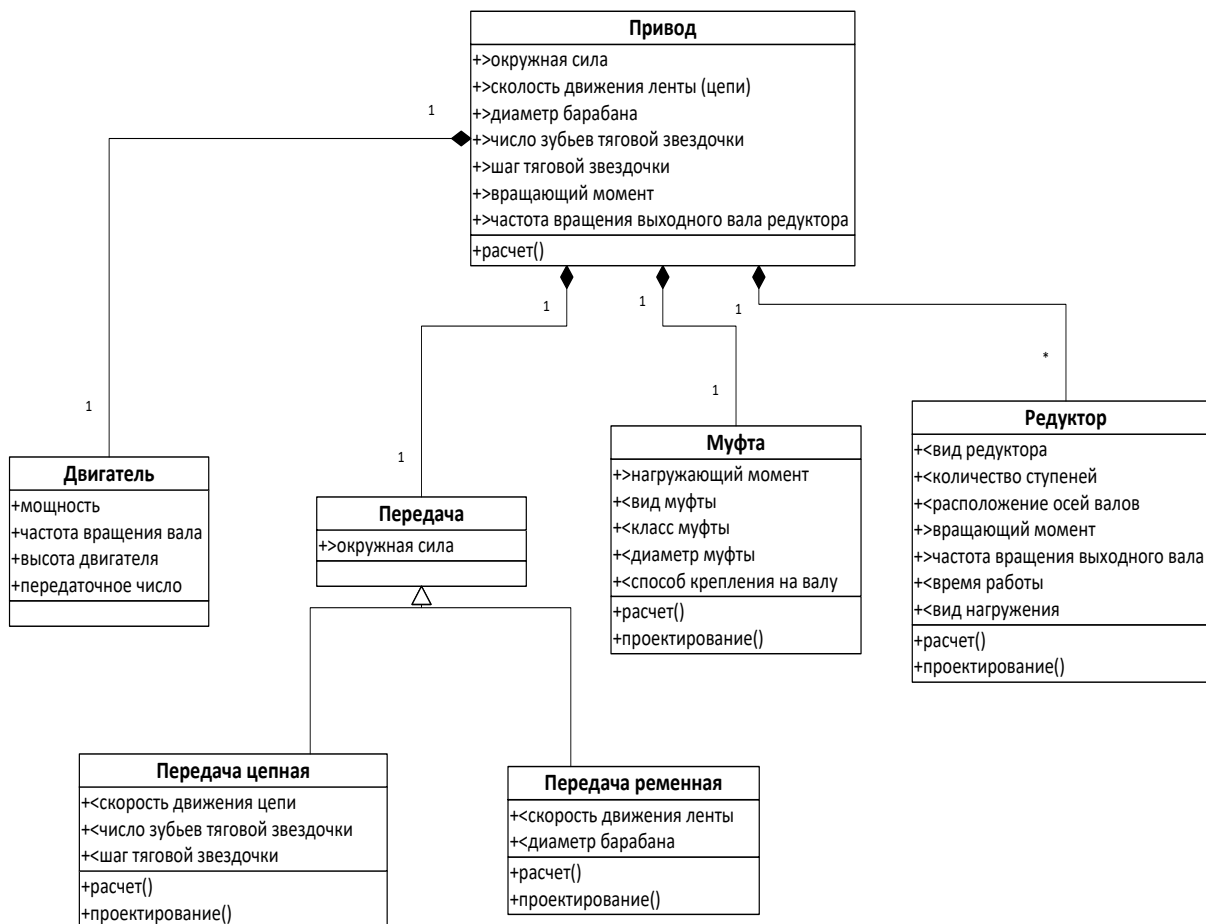
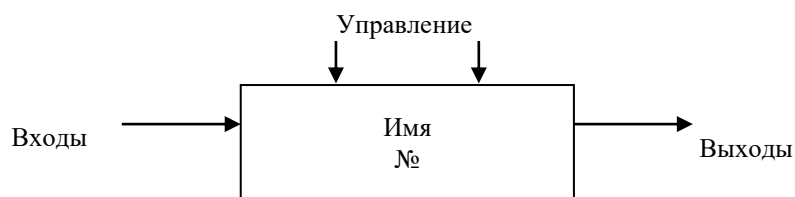


Рисунок 3 - Диаграмма классов объектов привода

Концептуально искусственный агент представляет собой объект-функцию. Модель функционального блока, связанного с агентом, определена стандартом IDEF0 (рисунок 4).



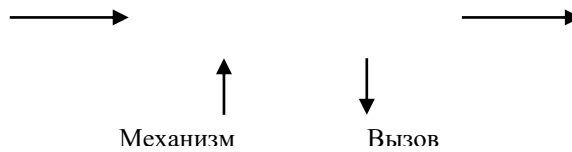


Рисунок 4 - Функциональный блок IDEF0

Важной частью объектного подхода является то, что объектно-ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы. Именно на это направлена многоагентная методология.

Модифицированная диаграмма классов для MAC содержит все компоненты функционального блока IDEF0.

На рисунке 5 приведена общая структура MAC проектирования. Она имеет три уровня: внешний, концептуальный и внутренний. На внешнем уровне отображаются внешние модели или группы описаний. Внутренний уровень содержит вычислительные программы, сгенерированные автоматически, исходя из информации внешнего уровня, а также протоколы решения задач в базах знаний.

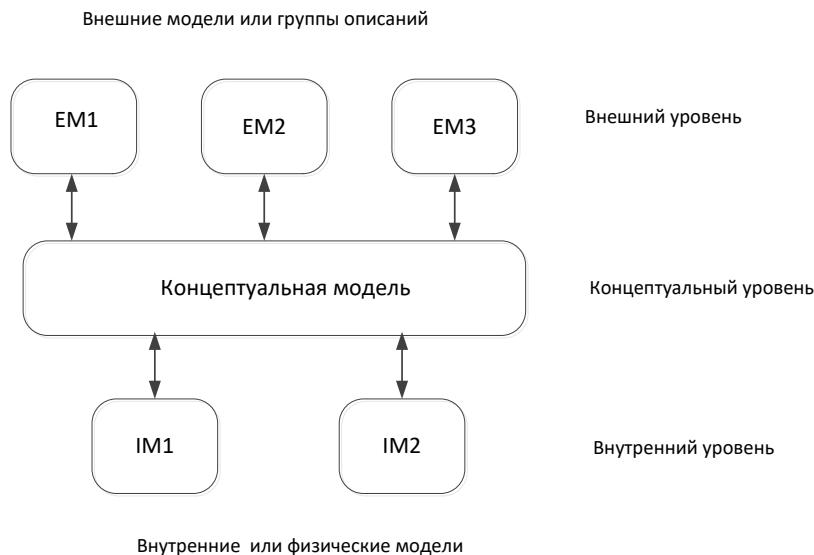


Рисунок 5 - Общая структура MAC проектирования

3 Технология экспертного программирования

При объектно-ориентированном подходе к проектированию программных систем есть две относительно самостоятельных и вместе с тем тесно взаимосвязанных стороны моделирования: статическая и динамическая. Статическое моделирование определяет структуру классов и объектов, а динамическое – их поведение. При разработке интеллектуальных систем проектирования с использованием модифицированного UML для статического моделирования используются диаграммы классов и объектов. Однако UML не располагает объектно-ориентированными средствами динамического моделирования, необходимыми для построения интеллектуальных систем.

Повышение производительности программирования может быть достигнуто на основе использования перспективных технологий, таких как *объектно-ориентированное программирование* (ООПр). Расширение числа специалистов, способных разрабатывать программные средства, может быть достигнуто за счёт привлечения носителей знаний в

прикладных областях, не владеющих программированием. Эта задача относится к компетенции другой области информатики – ИИ. В основу технологии ИИ заложено использование декларативных, то есть непроцедурных, не алгоритмических методов представления знаний. В этой связи ООПр представляет собой шаг в направлении сближения технологий программирования и ИИ. Экспертное программирование обеспечивает интеграцию лучших качеств обеих технологий.

В развитии процедурного программирования важнейшими этапами было создание технологий *структурного* и *модульного программирования*. Некоторые фундаментальные положения этих технологий используются в экспертном программировании.

В *структурном программировании* фундаментальной является теорема о структурировании, в которой доказано, что любая простая программа функционально эквивалентна структурной программе, составленной из элементов базисного множества (последовательность, если-то-иначе, делай-пока). Это означает, что любая программа, представляющая собой один функциональный узел, может быть составлена из структур трёх базовых типов.

Полезной в практическом плане рекомендацией структурного программирования является целесообразность декомпозиции сложных систем на части, содержащие не более 40...50 компонент. Эта рекомендация учитывается в экспертном программировании при определении максимального количества МЗ, входящих в один метод.

Следующим принципиально важным шагом в развитии технологии был переход к ООПр. Этот шаг позволяет разрабатывать эффективные программные средства за счёт лучших механизмов разделения данных, увеличения повторяемости кодов, использования стандартизованных интерфейсов пользователя и т.д.

Помимо графического представления системы в виде диаграммы классов агентов (рисунок 3) целесообразно иметь эквивалентное ему текстовое. Ниже приведено текстовое представление модуля расчёта передаточного числа для редуктора. Такое представление автоматически генерируется документатором системы экспертного программирования.

Знания в такой форме могут быть представлены на любом языке народов мира, а результат транслирования такого модуля при использовании соответствующих трансляторов - в форме подпрограммы на различных языках программирования.

МЗ: «PR3» - Расчёт передаточного числа

Предусловия запуска

имя	наименование	тип	условие
VidSE	Вид SE	STRING	Редуктор
nt	Частота вращения на выходе, об/мин	REAL	(0,)

Входные свойства

имя	наименование	тип	значение
<u>nh</u>	Частота вращения на входе, об/мин	REAL	
<u>nt</u>	Частота вращения на выходе, об/мин	REAL	

Механизм - Формула

$Urz = nh/nt$

Выходные свойства

имя	наименование	тип	значение
Urz	Передаточное число	REAL	

Приведённый МЗ соответствует внешнему уровню общей структуры МАС проектирования (рисунок 5) и относится к классу модулей типа формул. Этот модуль предназначен для расчёта передаточного числа и соответствует фразе: «Если вид сборочной

единицы – «редуктор» и частота на выходе более нуля, то на основе величин частот на входе и выходе можно подсчитать передаточное число по механизму-формуле».

С помощью МЗ типа формул можно формировать текстовые переменные. Константы заключаются в кавычки, например «Фрезеровать венец колеса m =». Числовые переменные преобразуются в текстовую форму с помощью функции Str(). Отдельные компоненты текста соединяются с помощью знака +. Если переменная m=1, а z=33, то модуль, приведённый ниже, сгенерирует текст: «Фрезеровать венец колеса m=1, z=33 окончательно».

МЗ: «FrSdPrC3» - Формирование содержания перехода окончательной обработки

Предусловия запуска

имя	наименование	тип	условие
ViZubKol\$	Вид зубчатого колеса	STRING	цилиндрическое, коническое
n_	Число одновременно обрабатываемых деталей	INTEGER	1
HarObr\$	Характер обработки	STRING	Черновая
StToch	Степень точности	INTEGER	10, 9

Входные свойства

имя	наименование	тип	значение
m_	Модуль детали	REAL	
z_	Число зубьев	INTEGER	

Механизм - Формула

SodPer\$ = "Фрезеровать венец колеса m = "+Str(m_)+", z = "+Str(z_)+"окончательно"

Выходные свойства

имя	наименование	тип	условие
SodPer\$	Содержание перехода	STRING	

Функциональные зависимости в методиках часто имеют табличную форму представления. Для ввода таких зависимостей в базы знаний используются модули с механизмами в виде таблиц. Прилагаемая к модулю таблица может иметь головку и боковик, которые в общем случае могут быть многоуровневыми. На представленном ниже модуле шапка на верхнем уровне содержит значения символьной переменной «Замена приспособлений», а на нижнем — диапазоны «Модуль детали». Боковик также имеет два яруса: на первом уровне левого столбца – значения переменной «Характеристика наладки», а на втором — переменной «Вид подачи червячной модульной фрезы». На основе этих двух входов по таблице можно определить значения выходной переменной — «Норматив подготовительно-заключительного времени базовый, мин». Таблица может быть недоопределённой, т.е. содержать пустые клетки. При значениях входных переменных, соответствующих этим клеткам, модуль не даст решения. Это означает ошибку в формировании базы знаний.

МЗ: «NzTrpbCh» - Назначение Тпз базового для червячных колес

Предусловия запуска

имя	наименование	тип	условие
ViZubKol\$	Вид зубчатого колеса	STRING	червячное

Входные свойства

имя	наименование	тип	значение
ZamPris\$	Замена приспособлений	STRING	
VidPod\$	Вид подачи червячной модульной фрезы	STRING	
HarNal\$	Характеристика наладки	STRING	
m_	Модуль детали	REAL	

Механизм - Таблица

Конфигурация свойств в таблице

	ZamPris\$
--	-----------

		m_
HarNal\$	VidPod\$	tpzb

Таблица

		с заменой установочных приспособлений			без замены установочных приспособлений		
		(0,6]	(6,12]	(12,)	(0,6]	(6,12]	(12,)
без замены фрезерного суппорта	радиальная	29	38	47	17	23	27
	тангенциальная	31	40	50	19	24	30
с заменой фрезерного суппорта	радиальная	39	52	67	27	37	42
	тангенциальная	41	56	72	29	40	52

Выходные свойства

имя	наименование	тип	значение
tpzb	Норматив подготовительно-заключительного времени базовый, мин	REAL	

Пример модуля выбора из БД. В этом модуле определяется тип редуктора и количество ступеней. Решение зависит от диапазона, в который попадает передаточное отношение редуктора, и взаимного расположения входной и выходной осей. Решение ищется по таблице «ТипРед», которая расположена в БД «DBPriv.SDB». Здесь определены логические условия доступа к таблице и приведено согласование имён полей таблицы и свойств из словаря БЗ.

МЗ: «PR4» - Выбор типа редуктора

Предусловия запуска

имя	наименование	тип	условие
-----	--------------	-----	---------

Входные свойства

имя	наименование	тип	значение
RaspOs\$	Расположение входной и выходной осей	STRING	
urz	Передаточное отношение редуктора	REAL	

Механизм - Выбор из БД

База данных:	каталог ресурсов\DBPriv.SDB
Таблица:	ТипРед

Условия доступа к таблице

	umin	<=	urz
AND	umax	>=	urz
AND	распосей	=	RaspOs\$

Согласование полей и свойств

ТипРед	=	TypRed\$
Nst	=	Nst

Выходные свойства

имя	наименование	тип	значение
Nst	Количество ступеней	INTEGER	
TypRed\$	Тип редуктора	STRING	

В интеллектуальных системах проектирования подобные модули действий (с точки зрения UML), или функциональные блоки (с точки зрения IDEF0), или правила-продукции (с точки зрения ИИ) принято называть МЗ. Эти модули представляются с помощью языка, который называется языком «деловой прозы». Он понятен для человека и в то же время необходим и достаточен для генерации программных средств на одном из языков программирования. Такие модули генерируются на основе словарей, для которых могут использоваться различные языки народов мира, а генерация соответствующих им подпрограмм может производиться на различных языках программирования.

Механизмы МЗ должны обеспечивать реализацию всех функций, которые могут потребоваться при формировании БЗ. В число таких функций входят следующие основные: вычисление по формулам (в том числе присвоение значений переменным), определение значений по таблицам, выбор значений из БД, обновление значений в БД, занесение значений в БД, вычисление значений с использованием подпрограмм, вычисление значений с помощью методов, сгенерированных из МЗ, вычисление значений с помощью исполняемых *exe*-модулей или *dll*-библиотек, сгенерированных другими системами.

Чтобы обеспечить возможность генерации 3D моделей с помощью БЗ, необходимо создать МЗ, с помощью которого на основе параметризованной в САД-системе модели детали и (или) сборочной единицы можно сгенерировать 3D модель изделия с рассчитанными в других модулях значениями размеров. Построенная 3D модель должна быть подключена к БЗ посредством специализированного МЗ. Интерфейс модуля генерации 3D модели в системе Sprut ExPro представлен на рисунке 6.

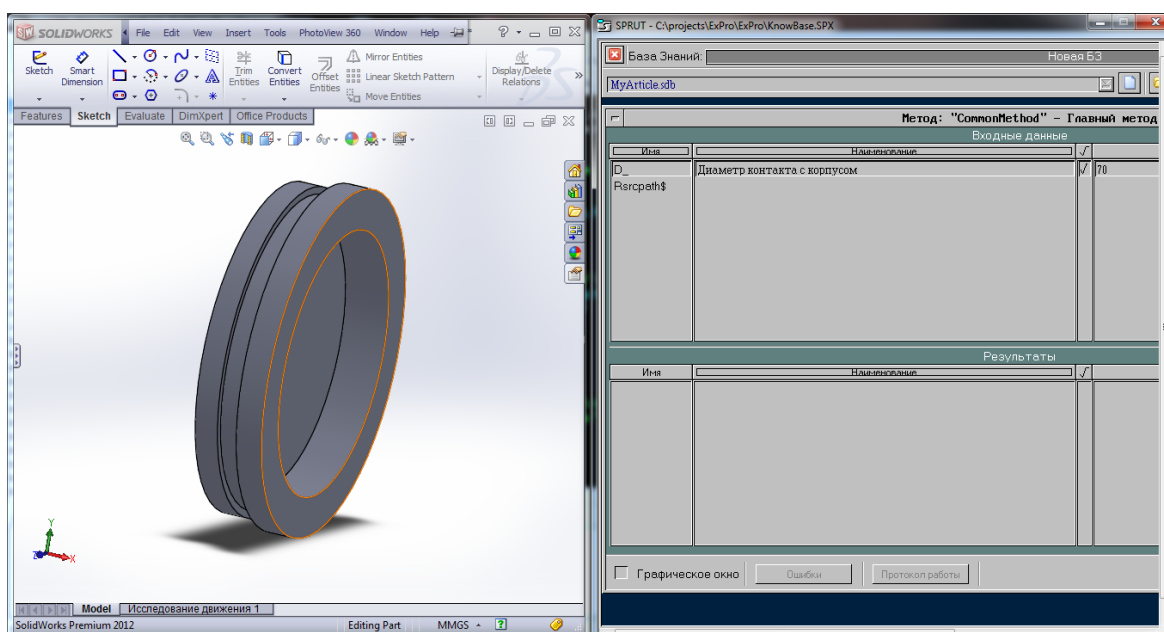


Рисунок 6 – Интерфейс генерации 3D модели в модуле знаний

Математический МЗ с идентификатором RsDP предназначен для расчёта дальности полета тела единичной массы, брошенного под углом к горизонту $Teta_0$ со скоростью V_0 . Скорость должна принимать значения больше нуля, а угол – в пределах от нуля до $\pi/2$.

МЗ: «RsDP» - Расчёт дальности полета

Предусловия запуска

имя	наименование	тип	условие
m	Масса, кг	REAL	1

Входные свойства

имя	наименование	тип	значение
V0	Скорость начальная, м/мин	REAL	(0,)
Teta0	D отверстия	REAL	(0,1.57)
g	Ускорение, м/сек^2	REAL	100

Механизм - DLL-модуль

Имя файла DLL-модуля: pol.dll

Согласование программных переменных и свойств МЗ

V0	вход	V0_
----	------	-----

Teta0	вход	Teta0
g	вход	g _
x	выход	x _
y	выход	y _

Выходные свойства

имя	наименование	тип	значение
x	Дальность полета, м	REAL	
y	Высота полета, м	REAL	

Современный инструмент моделирования должен позволять вводить и редактировать уравнения в математической форме. На рисунке 7 приведено окно редактора формул для представления системы уравнений движения тела единичной массы, брошенного под углом к горизонту, в скалярном виде на входном языке пакета MVS. Язык моделирования должен также позволять явно указывать набор искомым переменных. Например, в пакете MVS транслятор сам определяет допустимый набор искомым переменных, однако в ряде случаев этот набор может быть не единственным. Для систем уравнений с производными порядка выше первого необходимо иметь возможность явно указывать начальные значения «младших» производных (по умолчанию им присваивается нулевое начальное значение).

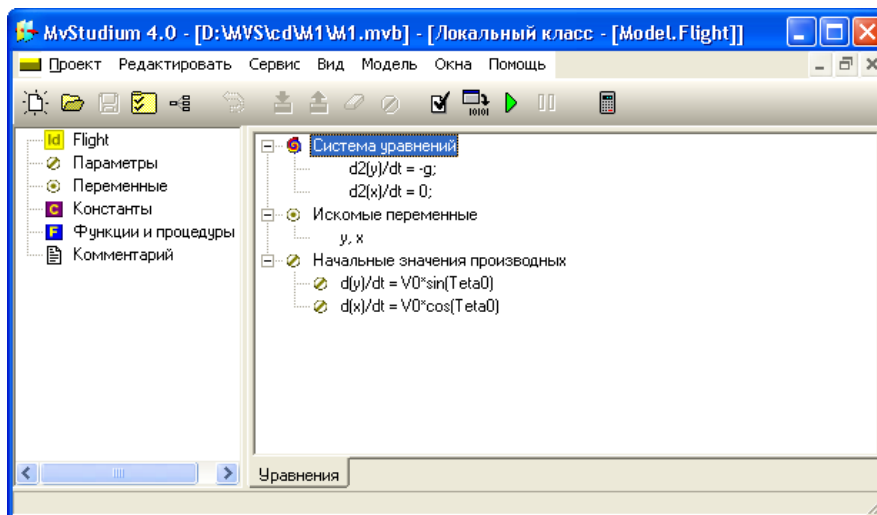


Рисунок 7 - Окно редактора формул пакета MVS

Неструктурированная совокупность МЗ в определённой прикладной области представляет собой БЗ этой области, аналогичную БЗ продукционной системы.

Модель структуры БЗ можно представлять как семантическую сеть взаимосвязанных модулей инженерных знаний. Поименованными узлами этой сети являются сами модули, а ориентированными рёбрами — входные и выходные переменные, наименования которых содержатся в словаре. Упомянутая сеть МЗ представляет собой внутреннюю модель, а полученные в результате трансляции подпрограммы — «физическую» модель.

В связи с ациклическостью всё множество МЗ может быть разбито на строго упорядоченное (ранжированное) множество подмножеств (рисунок 8).

При использовании технологии экспертного программирования процесс формирования МЗ, их трансляции с получением на одном из традиционных языков объектных или исполняемых модулей и тестирования производятся как одна операция. После получения необходимого набора МЗ производится генерация метода, использующего подмножество сгенерированных МЗ. Метод генерируется с использованием того же языка, что и МЗ, и представляет собой скомпилированную реализацию решателя для данного набора МЗ. В

скомпилированном механизме выбора продукционных правил используется схема управления, подобная сети Петри. Процесс планирования и разрешения конфликтов использует условия запуска МЗ. Программный модуль, соответствующий МЗ, выполняется, если имеются значения всех входных и управляющих переменных и выполнено условие его применения. Сгенерированный метод может быть использован в качестве механизма МЗ, что позволяет использовать иерархию правил.

Чтобы генерируемые методы согласно теореме о структурировании могли решать любые задачи программирования, в них должны быть реализованы три базовые структуры: следование, альтернатива и один из циклов. В экспертном программировании следование МЗ определяется последовательностью вычисления переменных, а альтернатива – условиями, заложенными в МЗ. Для формирования циклов введена выделенная переменная «конец цикла» (Fincalc), при появлении которой генерируется циклический метод, выполняющийся до тех пор, пока Fincalc, которой предварительно присвоено значение 0, не примет значение 1. Один метод может содержать только один цикл. Тело цикла включает модули рангов 1 и более. Модули ранга 0 обеспечивают начальные установки значений переменных и в цикле не исполняются. Возможность использования методов в качестве механизмов МЗ позволяет получать как последовательности циклов, так и программы с вложенными циклами.

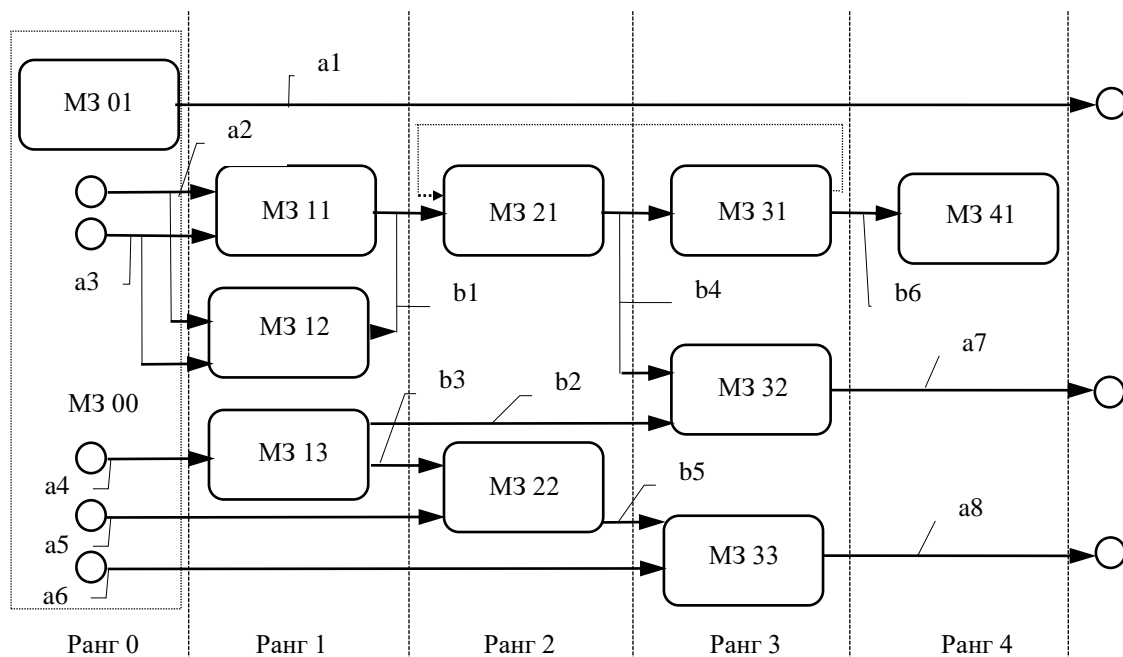


Рисунок 8 - Ранжированная семантическая сеть МЗ

Технология экспертного программирования использует методы продукционных систем и семантических сетей. В то же время ранжированная семантическая сеть МЗ в совокупности является МЗ, который может рассматриваться как фрейм. Итак, экспертное программирование интегрирует в себе все базовые методы представления знаний.

Заключение

В работе рассматривается совокупность двух миров: мир виртуальный, реализуемый ИнЗ, и мир реальный, реализуемый ИнВ. ИнЗ - это онтологический мир виртуальных агентов, представляющих объект-функции, способные генерировать новые данные на основе имеющейся информации. Предложены семантические вики — веб-приложения,

использующее машинообрабатываемые данные. Для дальнейшего расширения функциональности вики-систем целесообразно интегрировать их с технологиями ИИ. Создание «Экспертпедии» с использованием технологии экспертного программирования представляет собой попытку решения этой задачи, дополняя и развивая модели данных в машиностроении [17].

СПИСОК ИСТОЧНИКОВ

- [1] Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0. - <https://www.i-scoop.eu/industry-4-0>.
- [2] Industry 4.0: the Future of Smart Manufacturing - Praim - <https://www.praim.com> > Article.
- [3] *Gershenfeld, N.* The Internet of Things. Scientific American / N. Gershenfeld, R. Krikorian, D. Cohen / Oct. 2004.
- [4] *Evans, D.* The Internet of Things. How the Next Evolution of the Internet Is Changing Everything. Cisco White Paper. Cisco Systems (11 April 2011).
- [5] *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.
- [6] *Евгеньев, Г.Б.* Онтологическая методология создания интеллектуальных систем в машиностроении. //Известия высших учебных заведений. Машиностроение, №6, 2014, с.79- 86.
- [7] *Reynolds, J.C.* Theories of programming languages. Cambridge University Press. 1998, - 500 p. - <https://doi.org/10.1017/CBO9780511626364>.
- [8] *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. - СПб.: Питер, 2000. - 384 с.
- [9] *Helbig, H.* Knowledge Representation and the Semantics of Natural Language, Springer, Berlin, Heidelberg, New York, 2006.
- [10] *Евгеньев, Г.Б.* Sprut ExPro - средство генерации многоагентных систем проектирования в машиностроении. Части 1 и 2 // Известия высших учебных заведений. Машиностроение, 2017, №6, с.66-77; №7, с.60-71.
- [11] *Todd R. Weiss.* Wikipedia simplifies IT infrastructure by moving to one Linux vendor, Computerworld.com (October 9, 2008 (Computerworld)).
- [12] *Sikos, Leslie F.* (2017). Description Logics in Multimedia Reasoning. Cham: Springer International Publishing. doi:10.1007/978-3-319-54066-5. ISBN 978-3-319-54066-5.
- [13] *Колесов, Ю.Б.* Моделирование систем. Объектно-ориентированный подход / Ю.Б. Колесов, Ю.Б. Сениченков // BHV. 2016, с. 192.
- [14] *Skobelev, P.O.* Chapter 12. Multi-Agent Systems for Real Time Adaptive Resource Management. In book: Industrial Agents: Emerging Applications of Software Agents in Industry. Москва, Россия: Elsevier, 2015. 24 p.
- [15] *Скобелев, П.О.* Планирование сеансов связи между микроспутниками и сетью наземных станций с использованием мультиагентных технологий / П.О. Скобелев, А.Б. Иванов, Е.В. Симонова, В.С. Травин, А.А. Жиляев // Онтология проектирования. — 2014. — № 2(12). — С. 92-100.
- [16] *Евгеньев, Г.Б.* Основы автоматизации технологических процессов и производств. Т.1: Информационные модели. Т.2: Методы проектирования и управления - Москва, Изд-во МГТУ им. Н.Э. Баумана, 2015. - Т.1: - 441 с. - Т.2: - 479 с.
- [17] *Андрющенко, А.Н.* Тенденции и состояние в области управления справочными данными в машиностроении / А.Н. Андрющенко // Онтология проектирования. — 2012. — № 2(4). — С. 25-35.

EXPERTPEDIA AS A MEANS OF CREATION ONTOLOGICAL INTERNET OF KNOWLEDGE

G.B. Evgenev

*Moscow State Technical University named after N.E. Bauman, Moscow, Russia
g.evgenev@mail.ru*

Abstract

A methodology for creating semantic wiki systems for knowledge representation using artificial intelligence technologies is proposed. The methodology is based on multi-agent methods for creating knowledge bases and is suitable for developing design and management systems for digital intelligent productions. An Integrated structure of the Internet of

Knowledge and the Internet of Things has been developed. The Internet of knowledge is the ontological world of virtual agents that represent object functions that are capable of generating new data based on available information. Schema of the knowledge input into the computer has been analyzed. Semantic wiki system to create knowledge bases, which can be called "Expertpedia", is proposed on the basis of the technology of expert programming. The possibility of replacing the dictionary knowledge base allows us to provide implementation of the conceptual requirements of the wiki system: the use of any languages in the world. The knowledge base has the ability to add modules, select an analog module, translate the knowledge module into one of the programming languages, test the result, as well as determine the module's entry into the knowledge base and remove the selected module. These actions can be performed by any operator, which allows to implement the second conceptual basis of wiki systems: providing the possibility of independent replenishment and correction of the system content. Metaontology of Expertpedia have been developed. The methods of creating multi-agent knowledge bases for design and management in mechanical engineering are considered in detail. The architecture of the CAD agent is given. Examples of application of multi-agent systems for co-building of intelligent systems of semi-automatic design of engineering products are given. The technology of expert programming is described. To further expand the functionality of wiki systems, it is advisable to integrate them with artificial intelligence technologies.

Key words: *Industry 4.0, digital manufacturing, intelligent systems, intelligent systems engineering and manufacturing.*

Citation: *Evgenev GB. Expertpedia as a means of creation ontological Internet of knowledge [In Russian]. Ontology of designing. 2019; 9(3): 307-320. – DOI: 10.18287/2223-9537-2019-9-3-307-320.*

References

- [1] Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0. - <https://www.i-scoop.eu/industry-4-0>.
- [2] Industry 4.0: the Future of Smart Manufacturing - Praim - <https://www.praim.com> › Article.
- [3] **Gershenfeld N, Krikorian R, Cohen D.** The Internet of Things. Scientific American, Oct, 2004.
- [4] **Evans D.** The Internet of Things. How the Next Evolution of the Internet Is Changing Everything. Cisco White Paper. Cisco Systems (11 April 2011).
- [5] **Evgenev GB.** Industry 5.0 as an 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.
- [6] **Evgenev GB.** The ontological methodology for the creation of intelligent systems in mechanical engineering [In Russian]. // News of higher education institutions. Engineering. 2014; 6: 79-86.
- [7] **Reynolds JC.** Theories of programming languages. Cambridge University Press. 1998, - 500 p. - <https://doi.org/10.1017/CBO9780511626364>.
- [8] **Gavrilova TA, Khoroshevsky VF.** Knowledge Base of intellectual systems [In Russian]. SPb.: Peter, 2000. – 384 p.
- [9] **Helbig H.** Knowledge Representation and the Semantics of Natural Language, Springer, Berlin, Heidelberg, New York 2006.
- [10] **Evgenev GB.** Sprut ExPro is a tool for generating multi-agent design systems in mechanical engineering [In Russian]. Part 1 & 2 // News of higher educational institutions. Engineering. 2017; 6: 66-77 & 2017; 7: 60-71.
- [11] **Todd R. Weiss.** Wikipedia simplifies IT infrastructure by moving to one Linux vendor, Computerworld.com (October 9, 2008).
- [12] **Sikos LF.** Description Logics in Multimedia Reasoning. Cham: Springer International Publishing. 2017. - doi:10.1007/978-3-319-54066-5. ISBN 978-3-319-54066-5.
- [13] **Kolesov YB, Senichenkov YB.** Modeling systems [In Russian]. Object-oriented approach. BHV. 2016, p.192.
- [14] **Skobelev PO.** Chapter 12. Multi-Agent Systems for Real Time Adaptive Resource Management. In book: Industrial Agents: Emerging Applications of Software Agents in Industry. Москва, Россия: Elsevier, 2015. 24 p.
- [15] **Skobelev PO, Ivanov AB, Simonova EV, Travin VS, Jilyaev AA.** Planning communication sessions between microsatellites and a network of ground stations using multi-agent technologies [In Russian]. *Ontology of Designing.* 2014; 2(12): 92-100.
- [16] **Evgenev GB.** Fundamentals of automation of technological processes and production [In Russian]. Vol.1: Information models. Vol.2: Design and control methods - Moscow, Publishing house of MGTU N.E. Bauman, 2015. - Vol.1: 441 p. - Vol.2: 479 p.
- [17] **Andrichenko AN.** Tendencies and condition in the field of reference data management in the engineering industry [In Russian]. *Ontology of Designing.* 2012; 2(4): 25-35.

Сведения об авторе



Евгениев Георгий Борисович, 1938 г. рождения. Окончил Московский авиационный технологический институт в 1960 г., к.т.н. (1964), д.т.н. (1975). Лауреат премии Совета Министров СССР (1991). Профессор кафедры «Компьютерные системы автоматизации производства» МГТУ им. Н.Э. Баумана. Член Российской ассоциации искусственного интеллекта. В списке научных трудов более 150 работ в области САПР и ИИ.

Georgy Borisovich Evgenev (b. 1938) graduated from the Moscow aviation technological Institute in 1960, PhD (1964), Doctor of engineering (1975). He is Council of Ministers of the USSR prize winner (1991). Professor at Moscow State Technical University named after N.E. Bauman. Member of the Russian Association for Artificial Intelligence. Co-author of about 150 scientific articles in the field of CAD and AI.