

УДК 004.624

DOI 10.18698/0536-1044-2017-6-66-77

Sprut ExPro — средство генерации многоагентных систем проектирования в машиностроении.

Часть 1

Г.Б. Евгений

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

Sprut ExPro — A Means of Generating Multi-Agent Design Systems for Mechanical Engineering. Part 1

G.B. EvgenevBMSTU, 105005, Moscow, Russian Federation, 2nd Baumanskaya St., Bldg. 5, Block 1 e-mail: g.evgenev@mail.ru

i Трудоемкость и стоимость проектирования, как и качество его результатов, определяются объемом и глубиной инженерных знаний, заложенных в компьютер. Многоагентная технология компьютеризации инженерной деятельности позволяет специалисту, не обладающему глубокими познаниями в информатике, без помощи программистов создавать для себя и своих коллег специализированные рабочие места. При этом инженерная деятельность претерпевает качественные изменения: специалист вводит в компьютер данные технического задания и наблюдает за процессом генерации проекта, принимая принципиальные творческие решения путем выбора из вариантов, предлагаемых компьютером. Подобные системы с полным основанием можно отнести к принципиально новой категории полуавтоматических систем проектирования. Изложены теория и практика генерации многоагентных систем. Рассмотрена обобщенная модель класса искусственных агентов и введено их концептуальное определение. Приведен минимальный набор базовых характеристик искусственных агентов с раскрытием их содержания. Построена концептуальная блок-схема простого агента. Описана его реализация в системе Sprut ExPro.

Ключевые слова: искусственный интеллект, многоагентная методология, инструментальные программные средства, интеллектуальные системы проектирования.

i The complexity and cost of design, as well as the quality of its results, is determined by the volume and depth of the engineering knowledge, embodied in the computer. Multi-agent engineering computerization technology enables specialists, who do not have deep knowledge in computer science, to create specialized automated workplaces for themselves and their colleagues. In this case, the engineering work is undergoing a qualitative change: the designer inputs initial data into the computer and monitors the project generation process, making key creative decisions by choosing from the options offered by the computer. Such systems can justifiably be classed as an entirely new category of semi-automated design systems. A generalized model of the class of artificial agents is considered and a conceptual definition of the agents is introduced. The minimum set of basic characteristics of the artificial agents is presented and interpreted. A conceptual flow diagram of a simple agent is designed. The implementation of the agent in the Sprut ExPro system is described.

Keywords: artificial intelligence, multi-agent methodology, software tools, intelligent design systems.

В настоящее время в индустриально развитых странах начинается четвертая промышленная революция, призванная создать цифровые производства. Вследствие этого особую значимость приобретают работы, связанные с прогрессивными технологиями разработки программных средств для указанных производств. Цель работы — изложение такой новой технологии.

Основными концепциями дальнейшего развития систем автоматизации в машиностроении являются интеграция, интеллектуализация и индивидуализация [1, 2]. *Интеграция* призвана ликвидировать перекодировку информации при переходе от одной фазы жизненного цикла изделия к другой. В итоге будут формироваться все данные, необходимые для систем планирования и управления производством. *Интеллектуализация* должна сократить трудоемкость проектирования путем повышения уровня автоматизации систем и преобразования их из пассивного инструмента в руках специалиста в его активного партнера, обеспечивающего автоматическое принятие решений и генерацию, где это возможно, проектов изделий в целом или их узлов, а также технологических процессов их изготовления. *Индивидуализация* должна обеспечить преобразование систем автоматизации из обезличенного программного продукта в персональное программное средство, наполненное без помощи программистов индивидуальными знаниями экспертов.

Важнейшей методической основой для реализации перечисленных концепций является теория многоагентных систем (МАС) [3]. Многоагентным технологиям создания различных систем искусственного интеллекта уделяется значительное внимание за рубежом. В сентябре 2015 г. прошла 13-я Германская конференция, посвященная этой теме [4], имевшая цель развивать междисциплинарную связь между теорией и приложениями интеллектуальных агентов и МАС.

В статье [5] отмечено, что многоагентные программные средства традиционно рассматривают как промежуточные, позволяющие решать различные вспомогательные проблемы прикладных систем. В научных трудах [6–8] представлены многоагентные методы решения различных проблем в прикладных системах.

Данная работа принципиально отличается от упомянутых публикаций, так как в ней много-

агентная методология изложена как всеобъемлющий подход к созданию прикладных систем.

Концепция многоагентной методологии.

Обобщенная модель класса искусственных агентов (ИА) приведена на рис. 1 [3]. Любой ИА представляет собой открытую систему, помещенную в некоторую среду. В большинстве случаев этой средой является проект, формируемый в базах данных (БД).

Можно выделить три категории свойств ИА: импортируемые, экспортируемые и внутренние. Импортируемые свойства служат *рецепторами* ИА, формирующими его систему восприятия, экспортируемые — его *эффекторами*, функция которых заключается в воздействии на среду, т. е. на состояние проекта.

Свойства ИА всех трех категорий образуют его *память*, в которой хранится его текущее состояние.

Процессор ИА формирует 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. В целом процессор определяет *поведение* ИА. Поведение можно наблюдать, используя инспектор модели ИА, с помощью которого пользователь следит за состоянием свойств ИА, либо в графическом окне, в котором отображаются сгенерированные 3D-модели, чертежи и другая геометрическая информация.

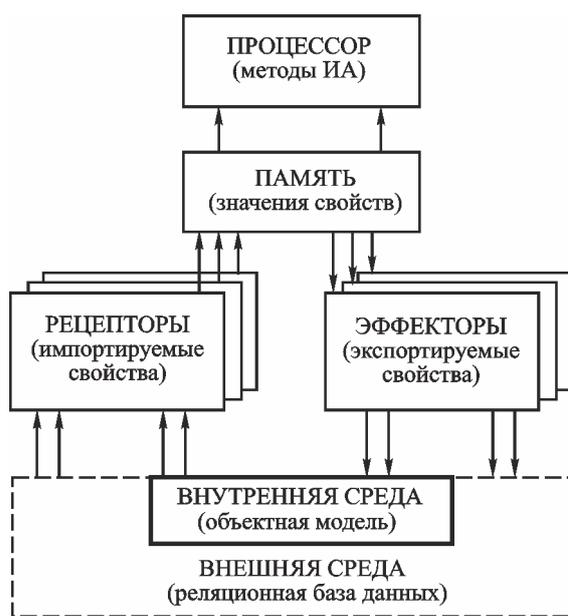


Рис. 1. Архитектура ИА

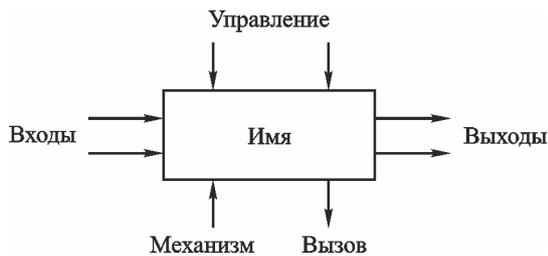


Рис. 2. Функциональный блок IDEF0

Концептуально ИА представляет собой объект-функцию. Модель функционального блока, связанного с ИА, определена стандартом IDEF0.

Внутри блока записывается его наименование, содержащее имя функции — глагол или глагольный оборот, описывающий действие, выполняемое блоком, а также существительное, определяющее предмет, на который направлено действие, и возможно дополнительная уточняющая информация (рис. 2). Это наименование должно быть именем ИА. Входы блока представляют собой объекты, которые необходимы для выполнения функции, и в результате ее выполнения преобразуются в выходы. Входы показывают все объекты, требующиеся для реализации функции, и она не может быть исполнена без получения этих объектов. Управление описывает условие, оказывающее влияние на выполнение функции, но само не подвергается расходованию или переработке.

К нижней части блока могут быть присоединены стрелки механизмов, обозначающие средство, обеспечивающее реализацию функции блока. Механизмы состоят из методов. Стрелки, направленные вверх, идентифицируют средства, поддерживающие выполнение функции, а направленные вниз служат стрелками вызова. Последние обозначают обращение из одной модели или из ее части к блоку, входящему в состав другой модели или ее части, обеспечивая их связь, т. е. разные модели или части одной и той же модели могут совместно использовать один и тот же элемент (блок).

Входы и выходы показывают, что делается функцией, управление — почему это делается, а механизмы — с помощью чего делается.

Выше были определены аспекты, связанные с функциональностью ИА. В то же время является объектом. Объект — это сущность, которой можно посылать сообщения и которая может на них реагировать, используя свои дан-

ные. Объект характеризуется состоянием и поведением. Состояние задается его параметрами, а его поведение зависит от связанной с ним функции. Функция (метод) определяет действия, которые может выполнить ИА. 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.

В минимальный набор базовых характеристик ИА должны входить [3]:

- активность, означающая способность к организации и реализации действий;
- реактивность или способность воспринимать состояние среды;
- автономность, т. е. относительная независимость, обеспечивающая собственное поведение на основе имеющихся у ИА данных и знаниях;
- общительность, вытекающая из необходимости решать свои задачи совместно с другими ИА;
- целенаправленность, предполагающая наличие собственных источников мотивации.

Рассмотрим применение МАС на примере интеллектуальных систем проектирования. В машиностроительных системах автоматизированного проектирования в качестве ИА выступают комплексы, сборочные единицы, детали и их элементы.

Активность ИА состоит в необходимости решения двух задач: структурного и параметрического синтеза. Первый заключается в выборе структуры подчиненных объектов, второй — в генерации значений собственных свойств, в результате чего из класса объектов, представленных в форме ИА, генерируется один экземпляр, который и включается в проект.

Реактивность ИА обеспечивает решение этих задач путем обмена информацией между ними непосредственно или через БД.

Автономность ИА основана на встроенных в них механизмах, где содержатся инженерные знания о различного рода расчетах, а также геометрические и графические знания в форме параметризованных моделей, обеспечивающих генерацию трехмерных образов и чертежей [9–14].

Общительность ИА имеет как вертикальную, так и горизонтальную составляющую. Вер-

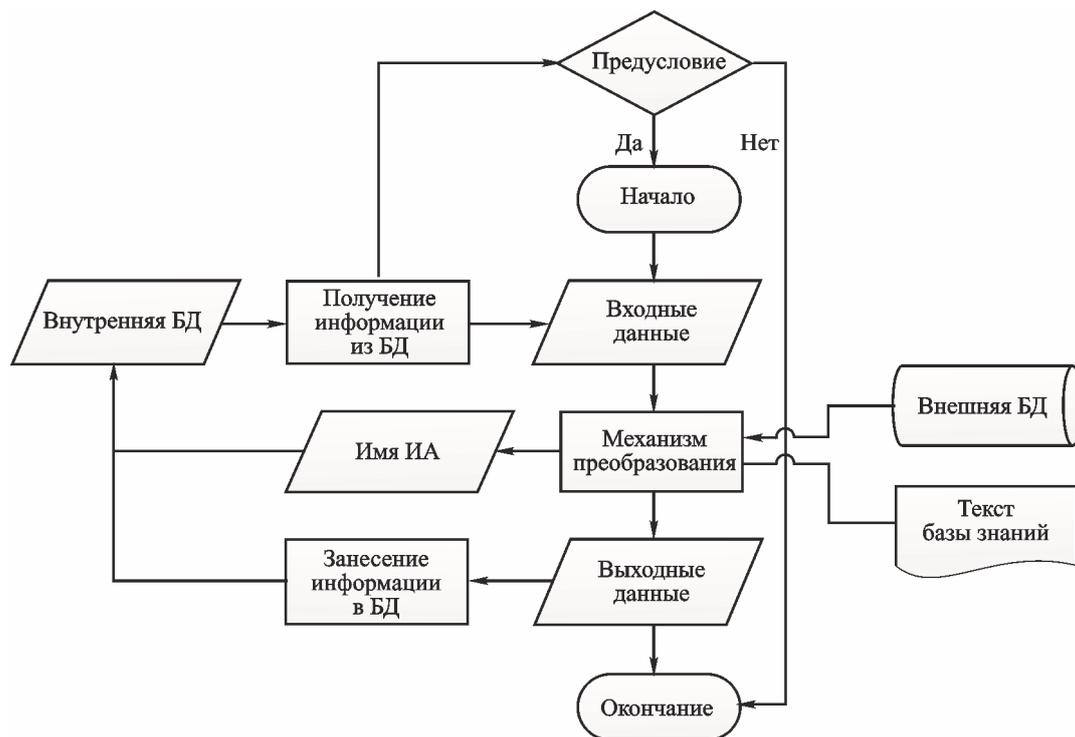


Рис. 3. Концептуальная блок-схема простого ИА

тикальная составляющая включает в себя обмен данными по иерархии «целое–часть» и «род–вид», а горизонтальная — обмен между конструктивно сопряженными, но не подчиненными друг другу по иерархии ИА.

Целенаправленность ИА определяется необходимостью реализации проекта, удовлетворяющего техническим требованиям заказчика, а также другим требованиям, предъявляемым разработчиком, в частности, связанным с поиском лучших решений.

Концептуальная блок-схема простого ИА. На рис. 3 представлена концептуальная блок-схема простого (атомарного) ИА, который не содержит в своем составе других ИА. В отличие от него комплексный (сборный) ИА использует для построения своего механизма множество других ИА.

Компоненты блок-схемы соответствуют описанным выше концептуальным положениям. Предусловие связано с управлением, входящим в модель IDEF0, и может быть явным или неявным. Неявное предусловие означает, что для выполнения ИА должны быть определены все управляющие и входные данные. Явное представление предусловия представляет собой предикат, содержащий логическое утверждение первого порядка об ИА. Здесь

предикат представляет собой в общем случае n -местную функцию, принимающую значение из множества $\{0, 1\}$ или {ложь (нет), истина (да)}. При значении «нет» ИА не выполняется, а при значении «да» подлежит выполнению, для чего необходимо получить информацию из БД, в которой хранятся значения переменных предиката и входных данных.

Входные данные преобразуются с помощью механизма с получением выходных данных. При выполнении преобразования формируется имя ИА, которое вместе со значениями выходных данных заносится в БД.

Таким образом, все приведенные на рис. 2 компоненты включены в блок-схему, что свидетельствует о том, что ИА представляет собой функциональный блок. Вместе с тем ИА соответствует концептуальным требованиям, предъявляемым к объектам, т. е. он является сущностью, которой можно посылать сообщения и которая реагирует на них, используя свои данные. Состояние ИА определяется значениями переменных предиката, входных и выходных данных, а поведение ИА — его механизмом преобразования.

Реализация простого ИА в системе Sprut Ex-Pro. На рис. 4 представлен интерфейс работы с БД в системе Sprut ExPro [9].

БД представляет собой общий словарь MAC, в которой по мере работы сохраняются результаты. Для каждой переменной БД включает в себя ее имя (идентификатор), используемое при генерации программной реализации MAC, наименование переменной с указанием при необходимости ее единицы измерения, тип переменной, а также имя ассоциативного списка для перечисляемых переменных.

На рис. 5 показано окно работы с ассоциативными списками, которые могут иметь как символьные, так и числовые переменные.

Простое предусловие ИА приведено на рис. 6, где он назван модулем инженерных знаний (МИЗ). Согласно этому предусловию,

ИА будет исполняться только для сплошных осей.

Каждый ИА имеет свой словарь (рис. 7) представляющий подмножество общего словаря и включающий в себя входные, выходные и управляющие переменные.

В системе Sprut ExPro ИА могут использовать различные механизмы, которые будут описаны далее. На рис. 8 приведено окно механизма ИА типа формул.

После того, как были определены все компоненты ИА и сгенерирована его программная реализация, можно провести тестирование (рис. 9), задавая входные данные и получая результаты.

Имя	Наименование	Тип	Асс. Список
Vid\$	Вид оси	STRING	VidAxle\$
Tip\$	Тип оси	STRING	TipAxle\$
d_	Диаметр оси, мм	REAL	
l_	Длина оси, мм	REAL	
c_	Ширина фаски, мм	REAL	
dr	Диаметр оси расчетный, мм	REAL	
li	Длина оси исходная, мм	REAL	
Mi	Момент изгибающий, Нмм	REAL	

Рис. 4. База данных MAC

Имя	Наименование	Тип	Значения Ассоциативного Списка
VidAxle\$	Виды осей	STRING	гладкие
TipAxle\$	Типы осей	STRING	с буртиком
KonAxle\$	Конфигурации осей	STRING	
MarMat\$	Марки стали	STRING	
VidTer\$	Виды термообработки	STRING	

Рис. 5. Ассоциативные списки переменных перечисляемого типа

Имя	Наименование	Условие
Kon\$	Конфигурация оси	✓ сплошная

Рис. 6. Предусловие ИА

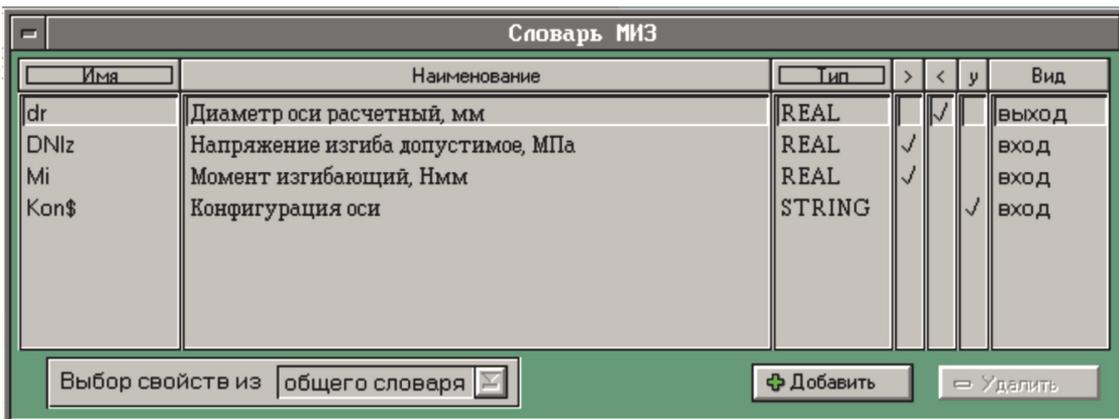


Рис. 7. Словарь ИА

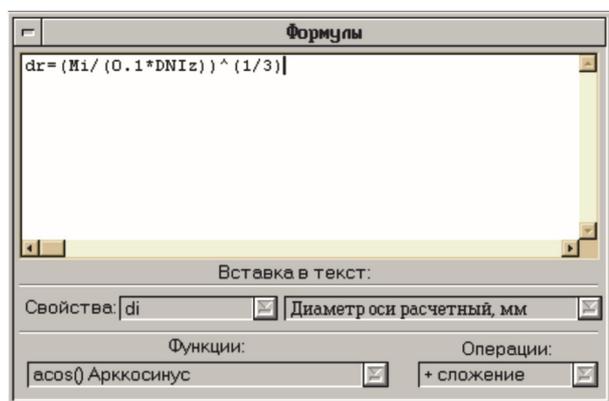


Рис. 8. Механизм ИА типа формул

Выше был представлен простейший механизм функционирования ИА — расчет по формулам (в том числе присвоение значений пере-

менным). Чтобы МАС была способна решать все множество задач в различных областях, необходимо обеспечить возможность расширения механизмов. В их число должны войти:

- определение значений по таблицам;
- выбор значений из БД;
- обновление значений в БД;
- занесение значений в БД;
- генерация 3D-моделей;
- вычисление значений с использованием:
 - подпрограмм;
 - методов, сгенерированных из модулей инженерных знаний;
 - exe-модулей или dll-библиотек, созданных другими системами.

В системе Sprut ExPro с помощью модулей знаний (МЗ) типа формул можно создавать

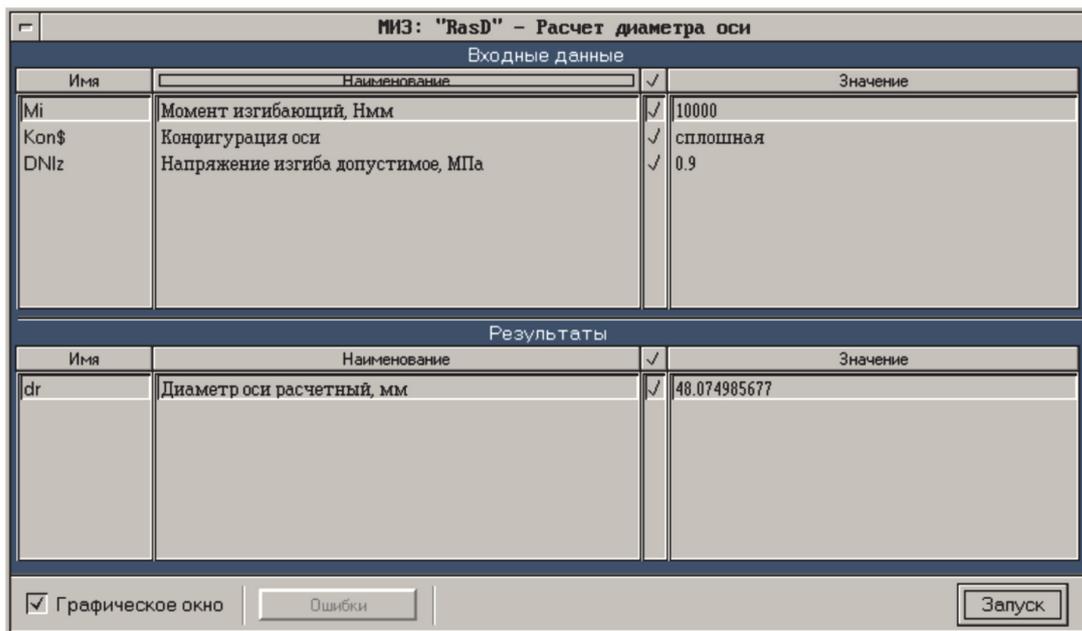


Рис. 9. Моделирование работы ИА

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, мм	REAL	
z_	Число зубьев	INTEGER	

Механизм — Формула

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

Выходные свойства

имя	наименование	тип	значение
SodPer\$	Содержание перехода	STRING	

Рис. 10. Символьное представление объекта-функции формирования текстовой переменной

МЗ: "NzTpzbCh" — 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, мм	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	

Рис. 11. Символьное представление табличного объекта-функции с ярусными шапкой и боковином

чительного времени для операций обработки червячных колес. Для решения этой задачи необходимо задать вид замены приспособлений (с заменой или без замены установочных приспособлений), вид подачи червячной модульной фрезы (радиальная или тангенциальная), характеристику наладки (без замены или с заменой фрезерного суппорта), а также модуль зубчатого венца. В итоге будет получен базовый

норматив подготовительно-заключительного времени в минутах.

Важнейшей функцией МАС является работа с внешними БД. Пример модуля выбора из БД приведен на рис. 12. 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. Реше-

umin	umax	расposesей	ТипРед	Nst
6.3	50	параллельное	цилиндрический	2
16	250	параллельное	цилиндрический	3

Рис. 13. Экранная форма выбора одного решения из БД

МИЗ: "ZnDnZbVn" — Занесение данных зубчатого венца

Предусловия запуска

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

Входные свойства

имя	наименование	тип	значение
beta	Угол наклона зуба, град	REAL	
StToch	Степень точности	INTEGER	
Null	Ноль	REAL	
VidZub\$	Вид зуба	STRING	
z_	Число зубьев	INTEGER	
Lr	Расчетная длина обрабатываемой поверхности, мм	REAL	
ViZubKol\$	Вид зубчатого колеса	STRING	
m_	Модуль детали, мм	REAL	

Механизм — Обновить в БД

База данных: [каталог ресурсов\DB.SDB](#)

Таблица: DanZbVn

Условия доступа к таблице

No=Null

Согласование полей и свойств

m_	=	m_
z_	=	z_
StToch	=	StToch
beta	=	beta
ViZubKol	=	ViZubKol\$
L	=	Lr
VidZub	=	VidZub\$

Рис. 14. Символьное представление объекта-функции обновления БД

ние ищут по таблице «ТипРед», которая расположена в БД «DBPriv.SDB». На рис. 12 определены логические условия доступа к таблице и приведено согласование имен ее полей и свойств из словаря базы знаний. Подобные модули реализуют функции рецепторов (см. рис. 1).

Из БД могут быть выбраны одна или несколько записей, соответствующих условию. В первом случае процесс проектирования идет автоматически, а во втором отобранная информация выводится на экран и пользователь проводит окончательный выбор решения (рис. 13).

Аналогичный модуль обновления БД приведен на рис. 14. Здесь данные заносятся в ячейку с фиксированным значением поля «No», равно нулю. Подобные модули реализуют функции эффекторов (см. рис. 1).

Модули знаний с механизмом типа Prt-подпрограммы чаще всего используются для выполнения действий, связанных с вводом и выводом информации. Эти подпрограммы генерируются с помощью инструментальных программных средств, которые будут описаны далее. На рис. 15 приведен пример модуля ввода символьных данных. Этот модуль начинает действовать, если зубчатое колесо принадлежит к виду червячных. Входные переменные содержат

МЗ: "VvDCher" — Ввод данных колеса червячного Предусловия запуска

имя	наименование	тип	условие
<u>ViZubKol\$</u>	Вид зубчатого колеса	STRING	червячное

Входные свойства

имя	наименование	тип	значение
<u>RsrcPath\$</u>		STRING	

Механизм — Prt-модуль

Имя файла PRT-модуля: [каталог ресурсов\VvDCher.prt](#)

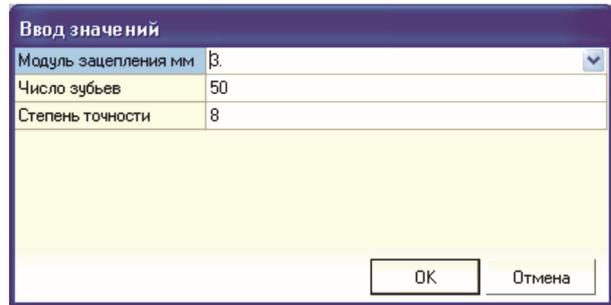
Согласование PRT-переменных и свойств МИЗ

<u>m_</u>	выход	<u>m_</u>
<u>z_</u>	выход	<u>z_</u>
<u>StToch</u>	выход	<u>StToch</u>
<u>NameMIZ\$=</u>		VvDCher
<u>RsrcPath\$</u>	вход	<u>RsrcPath\$</u>
<u>exit_</u>	выход	<u>exit_</u>

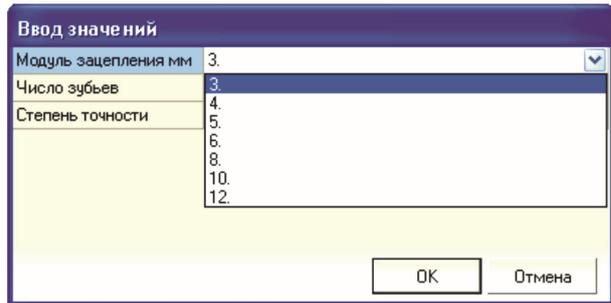
Выходные свойства

имя	наименование	тип	значение
<u>z_</u>	Число зубьев	INTEGER	30
<u>exit_</u>	Выход	REAL	0
<u>m_</u>	Модуль детали, мм	REAL	3
<u>StToch</u>	Степень точности	INTEGER	8

Рис. 15. Символьное представление объекта-функции ввода данных



а



б

Рис. 16. Экранные формы ввода данных

служебную переменную, обеспечивающую автоматическое согласование путей. Инструментальное средство экспертного программирования генерирует эту переменную автоматически. Как и в случае работы с БД, здесь проводится согласование переменных Prt-подпрограммы и словаря базы знаний. При необходимости могут использоваться константы, например, с заданием имени модуля (NameMIZ\$= VvDCher).

На рис. 16 представлены экранные формы, с помощью которых осуществляется ввод данных. На рис. 16, а показано итоговое состояние формы. Результаты можно принять, нажав кнопку «OK», или отменить, прервав дальнейшее выполнение программы. При этом проводится изменение выходной переменной «exit» (см. рис. 15), с помощью которой выполняется управление процессом. Если эта переменная равна нулю, то процесс продолжается, а если единице, то прерывается.

Переменные перечисляемого типа на экранной форме отмечены цветом. Для выбора значения из ассоциативного списка необходимо нажать на кнопку «OK» (рис. 16, б).

Данная статья содержит информацию, связанную с созданием MAC, включая реализацию простого ИА. Описание ИА, способных генерировать 3D-модели машиностроительных изделий, комплексных ИА и методов интеграции про-

граммных комплексов с использованием MAC будут рассмотрены во второй части работы.

Выводы

1. Разработана концепция создания прикладных многоагентных систем. Предложена интерпретация ИА как объектов-функций.

2. Представлена интерпретация базовых характеристик ИА применительно к интеллектуальным системам проектирования в машиностроении.

3. Построена концептуальная блок-схема простого ИА.

4. Рассмотрена реализация простого ИА в системе Sprut ExPro.

Литература

- [1] Евгеньев Г.Б. *Интеллектуальные системы проектирования*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2012. 420 с.
- [2] Евгеньев Г.Б., ред. *Основы автоматизации технологических процессов и производств*. В 2 т. Москва, Изд-во МГТУ им. Н. Э. Баумана, 2015. Т. 1. 441 с., т. 2. 479 с.
- [3] Тарасов В.Б. *От многоагентных систем к интеллектуальным организациям: философия, психология, информатика*. Москва, Эдиториал УРСС, 2002. 352 с.
- [4] Müller J.P., Ketter W., Kaminka G., Wagner G., Bulling N. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, Revised Selected Papers, Springer, 2015, 291 p.
- [5] Braubach L., Pokar A., Kalinowski J., Jander K. Tailoring Agent Platforms with Software Product Lines. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 3–21.
- [6] Hrabia C-E., Masuch N., Albayrak S. Autonomy in Complex Systems. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 22–41.
- [7] Diaconescu I.M., Wagner G. Modeling and Simulation of Web-of-Things Systems as Multi-Agent Systems. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, p. 248.
- [8] Bender J., Kehl S., Muller J.P. A Comparison of Agent-Based Coordination Architecture Variants for Automotive Product Change Management. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 137–153.
- [9] СПРУТ-Технология. URL: www.sprut.ru (дата обращения 15 декабря 2016).
- [10] Евгеньев Г.Б., Кокорев А.А., Пиримяшкин М.В. Интеллектуальные системы полуавтоматического проектирования и быстрого прототипирования изделий машиностроения. *Евразийский Союз Ученых. Технические науки*, 2015, № 9(18), с. 19–25. URL: <http://euroasia-science.ru/tehnicheskie-nauki/intellektualnye-sistemy-poluavtomaticheskogo-proektirovaniya-i-bystrogo-prototipirovaniya-izdelij-mashinostroeniya/> (дата обращения 15 января 2017).
- [11] Евгеньев Г.Б., Кокорев А.А., Пиримяшкин М.В. Метод создания геометрических баз знаний. *Инженерный вестник*, 2016, № 1, с. 1201–1218. URL: <http://engsi.ru/doc/832611.html> (дата обращения 20 января 2017).
- [12] Евгеньев Г.Б., Кокорев А.А., Пиримяшкин М.В. Разработка интеллектуальной системы трехмерного проектирования деталей. Часть 2. *Инженерный вестник*, 2016, № 2, с. 520–534. URL: <http://engsi.ru/doc/834324.html> (дата обращения 20 января 2017).
- [13] Колесов Ю.Б. *Объектно-ориентированное моделирование сложных динамических систем*. Санкт-Петербург, Изд-во СПбГПУ, 2004. 239 с.
- [14] Евгеньев Г.Б. Полуавтоматические системы типового вариантного проектирования. *Актуальные проблемы технических наук в России и за рубежом. Сб. науч. тр. междунар. науч.-практ. конф.*, Новосибирск, Инновационный центр развития образования и науки, 2016, № 3, с. 9–14.

References

- [1] Evgenyev G.B. *Intellektual'nye sistemy proektirovaniia* [Intelligent systems design]. Moscow, Bauman Press publ., 2012. 420 p.

- [2] *Osnovy avtomatizatsii tekhnologicheskikh protsessov i proizvodstv* [Basics of automation of technological processes and production]. Vol. 2. Ed. Evgenev G.B. Moscow, Bauman Press, 2015. Vol. 1. 441 p. Vol. 2. 479 p.
- [3] Tarasov V.B. *Ot mnogoagentnykh sistem k intellektual'nym organizatsiiam: filozofii, psikhologii, informatika* [From multiagent systems to intellectual organizations: philosophy, psychology, computer science]. Moscow, Editorial URSS publ., 2002. 352 p.
- [4] Müller J.P., Ketter W., Kaminka G., Wagner G., Bulling N. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, Revised Selected Papers, Springer, 2015, 291 p.
- [5] Braubach L., Pokar A., Kalinowski J., Jander K. Tailoring Agent Platforms with Software Product Lines. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 3–21.
- [6] Hrabia C-E., Masuch N., Albayrak S. Autonomy in Complex Systems. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 22–41.
- [7] Diaconescu I.M., Wagner G. Modeling and Simulation of Web-of-Things Systems as Multi-Agent Systems. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, p. 248.
- [8] Bender J., Kehl S., Muller J.P. A Comparison of Agent-Based Coordination Architecture Variants for Automotive Product Change Management. Multiagent System Technologies. *13th German Conference, MATES 2015, Cottbus, Germany, 28–30 September 2015*, pp. 137–153.
- [9] *SPRUT-Tekhnologiya* [SPRUT-Technology]. Available at: www.sprut.ru (accessed 15 December 2016).
- [10] Evgenev G.B., Kokorev A.A., Piriashkin M.V. Intellektual'nye sistemy poluavtomaticheskogo proektirovaniia i bystrogo prototipirovaniia izdelii mashinostroeniia [Semi-intelligent systems design and rapid prototyping of engineering products]. *Evrasiiskii Soiuz Uchenykh. Tekhnicheskie nauki* [Eurasian Union of Scientists. Engineering science]. 2015, no. 9(18), pp. 19–25. Available at: <http://euroasia-science.ru/tehnicheskie-nauki/intellektualnye-sistemy-poluavtomaticheskogo-proektirovaniia-i-bystrogo-prototipirovaniia-izdelij-mashinostroeniia/> (accessed 15 January 2017).
- [11] Evgenev G.B., Kokorev A.A., Piriashkin M.V. Metod sozdaniia geometricheskikh baz znaniia [A method of creating geometric knowledge bases]. *Inzhenernyi vestnik* [Engineering journal]. 2016, no. 1, pp. 1201–1218. Available at: <http://engsi.ru/doc/832611.html> (accessed 20 January 2017).
- [12] Evgenev G.B., Kokorev A.A., Piriashkin M.V. Razrabotka intellektual'noi sistemy trekhmernogo proektirovaniia detalei. Chast' 2 [The development of intelligent system of three-dimensional design details. Part 2]. *Inzhenernyi vestnik* [Engineering journal]. 2016, no. 2, pp. 520–534. Available at: <http://engsi.ru/doc/834324.html> (accessed 20 January 2017).
- [13] Kolesov Iu.B. *Ob'ektno-orientirovannoe modelirovanie slozhnykh dinamicheskikh sistem* [Object-oriented modeling of complex dynamic systems]. Sankt-Petersburg, SPbSPU publ., 2004. 239 p.
- [14] Evgenev G.B. Poluavtomaticheskie sistemy tipovogo variantnogo proektirovaniia [Semi-automatic model trial design]. *Aktual'nye problemy tekhnicheskikh nauk v Rossii i za rubezhom. Sb. nauchnykh trudov po itogam mezhdunarodnoi nauchno-prakticheskoi konferentsii* [Actual problems of technical Sciences in Russia and abroad. Collection of scientific papers based on the results of international scientific-practical conference]. Novosibirsk, Innovatsionnyi tsentr razvitiia obrazovaniia i nauki publ., 2016, no. 3, pp. 9–14.

Статья поступила в редакцию 22.02.2017

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) — Doctor of Science (Eng.), Professor, Department of Computer Systems for Industrial Automation. Bauman Moscow State Technical University (105005, Moscow, Russian Federation, 2nd Bauman-skaya St., Bldg. 5, Block 1, e-mail: g.evgenev@mail.ru).