

Архитектура мультиагентной платформы для решения задач структурно-параметрического синтеза объектов. Многоаспектное моделирование

Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ"

Показана необходимость дополнения агентной архитектуры распределенных САПР многоаспектной моделью объекта разработки, позволяющей повысить гибкость представления инженерных знаний и управления процессом проектирования. Обоснована возможность перехода при автоматизированном проектировании от жесткой последовательности стадий и этапов к параллельному выполнению проектных процедур.

Ключевые слова: многоаспектная модель, структурный синтез, параметрический синтез, база знаний, программный агент, агентная сеть.

Проектирование сложных технических объектов требует одновременного рассмотрения большого числа отношений различных видов (причинно-следственных, временных, пространственных) между их элементами и свойствами и процессов различной физической природы (механических, электрических, гидравлических и др.). Построение единой информационной или математической модели таких объектов практически невозможно, поскольку для описания различных видов отношений и связей требуется разный математический аппарат – различные методы континуальной и дискретной математики. Принцип декомпозиции приводит к формированию вместо единой модели некоторого набора субмоделей (частных моделей), каждая из которых отражает определенный аспект рассмотрения, т.е. частную точку зрения на объект проектирования. Количество и виды этих субмоделей могут изменяться при переходе между иерархическими уровнями (изделие, агрегат, узел, деталь) и стадиями разработки. Для самолета важнейшими из таких субмоделей служат геометрическая, весовая, аэродинамическая модели, модели динамики полета, силовой установки, компоновки и центровки, эффективностные и экономические модели [1]. Заметим, что перечисленные модели относятся только к функциональному аспекту, а при задачах ресурсного (прочностного), конструктивного и технологического проектирования к общему перечню добавляются соответствующие субмодели.

Однако разделение описания сложного объекта на частные модели и соответствующие группы задач, существенно упрощая процесс моделирования в рамках каждого аспекта, столь же существенно усложняет процедуры согласования частных проектных решений, полученных в этих рамках. Декомпозиция технической системы понижает явную сложность, но повышает так называемую неявную сложность, связанную с трудностями определения ожидаемых свойств системы по характеристикам ее элементов, что является проявлением свойства эмерджентности сложных систем [2].

Весь состав подлежащих решению задач может быть классифицирован по таким признакам, как иерархический уровень, аспект и тип задачи (см. рис. 1). На рисунке выделена область, примерно соответствующая задачам формирования

облика изделия. В табл. 1 в качестве примера приведен перечень задач проектирования для одного из агрегатов – крыла.

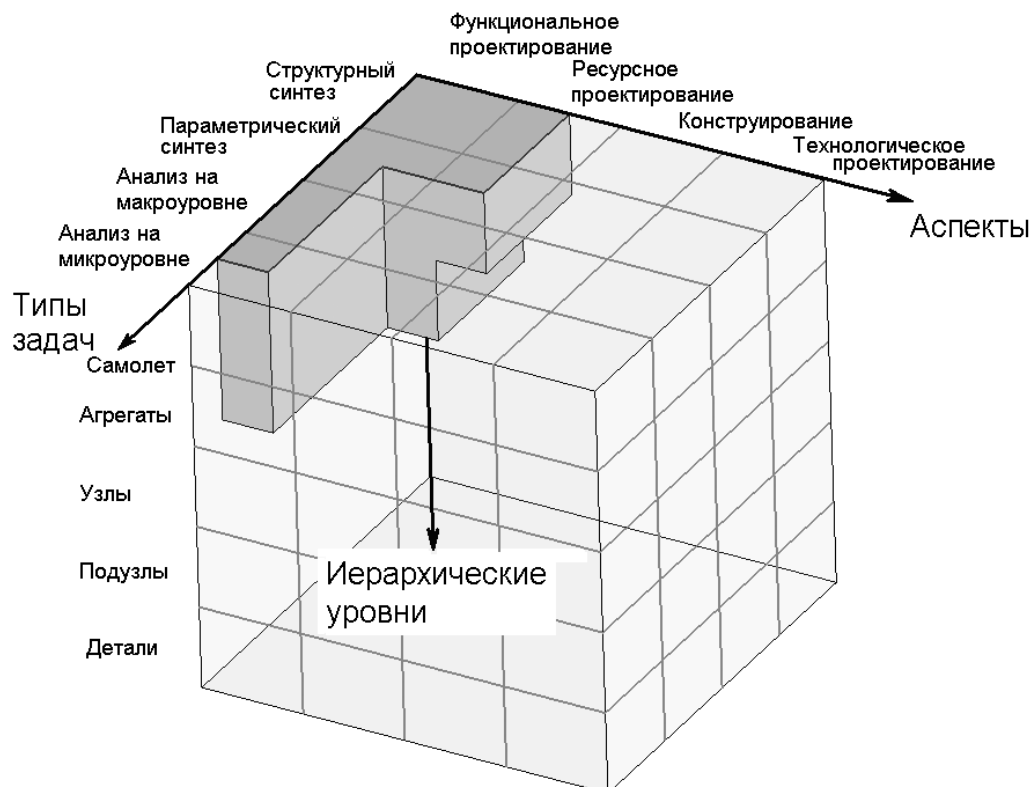


Рис. 1. Классификация задач проектирования

По каждой оси «системного куба» (иерархические уровни, типы задач, аспекты) допустимо перемещение только в одном направлении, но последовательность отдельных шагов по различным осям не ограничена ничем, кроме наличия необходимых исходных данных, и нигде не оговорено, в каком направлении следует делать первые шаги, при каких условиях изменять направление движения и до каких пор вообще допустимо движение в одном направлении. Кроме того, мы встретимся со «сцепленными» (связанными) задачами, множества переменных которых пересекаются, как, например, в задачах аэроупругости.

Это значит, что стратегия проектирования может быть вовсе не так жестко регламентирована, как при последовательном «поаспектном» проходе. Однако для этого нужна иная организация информационных и процедурных компонент ПО САПР.

В частности, гибкое управление последовательностью проектных процедур, близкое к адаптивной стратегии проектирования, могла бы обеспечить *многоаспектная модель объекта проектирования*. Одной из современных тенденций развития САПР как раз и является стремление разработчиков к построению систем взаимосвязанных моделей, характеризующих различные аспекты описания объекта проектирования.

Таблица 1

Задачи проектирования самолета на уровне агрегата «Крыло»

Аспекты	Типы задач			
	Структурный синтез	Параметрический синтез	Анализ на макроуровне	Анализ на микроуровне
Функциональное проектирование	Определение формы крыла, типов и расположения средств механизации	Расчет геометрии крыла и его частей (площадь, размеры)	Расчет аэродинамических характеристик (C_x , C_y и др.)	Численное моделирование обтекания крыла
Ресурсное проектирование	Определение конструктивно-силовой схемы крыла (типов и числа силовых элементов)	Проектировочный расчет размеров силовых элементов	Кинематический и динамический анализ подвижных элементов	Численное моделирование НДС элементов крыла
Конструирование	Компоновка элементов конструкции и оборудования (топливо, приводы механизации, шасси и др.)	Распределение компоновочных элементов по сечениям	Расчет центровки	—
Технологическое проектирование	Выбор схемы технологического членения и сборки	Расчет сборочных размерных цепей	Оценка технологичности конструкции (design for assembly)	—
	Выбор схемы листовой штамповки (вытяжка, обтяжка и др.), числа переходов и др.	Расчет размеров заготовки, усилий вытяжки, давления прижима и др.	Поверочные расчеты усилий, коэффициента вытяжки, точности и др.	Численное моделирование процесса формообразования и НДС заготовки

Примером такого подхода может служить работа [3], где объединены модули весового проектирования, аэродинамического и прочностного расчетов. К этому направлению относятся и работы ряда зарубежных исследовательских групп, в частности Делфтского Технического университета (Нидерланды) [4], где создана знаниеориентированная система Multi-Model Generator (MMG), схема

которой показана на рис. 2. Система использует средства языка GDL (General-purpose Declarative Language). Совмещенная информационная модель объекта охватывает аэродинамический, прочностной и производственно-экономический «слой».

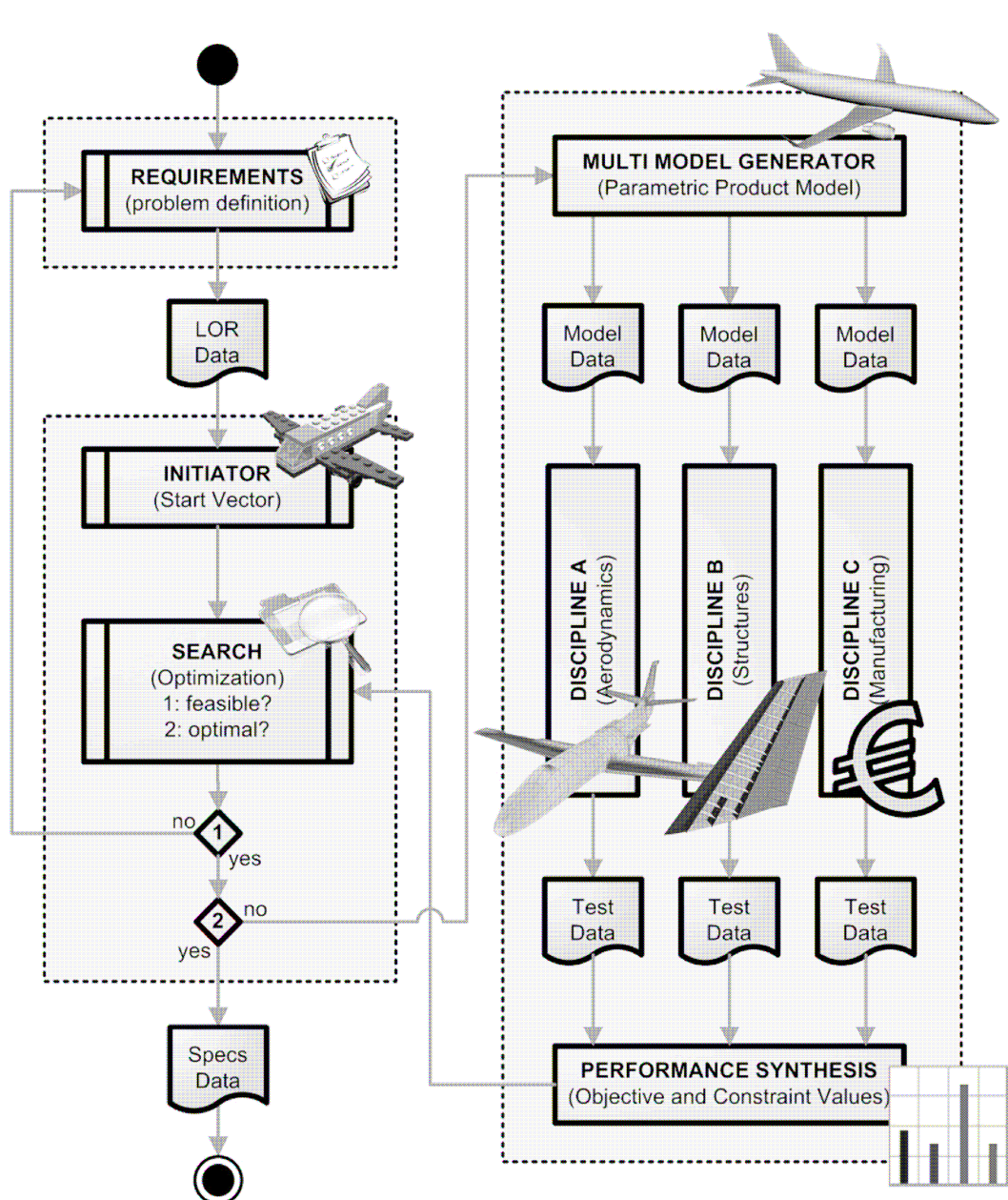


Рис. 2. Схема знаниеориентированной системы Multi-Model Generator

К настоящему времени разработан ряд теорий, связанных с решением проблемы многоаспектного моделирования, истоки которых можно видеть еще в Общей Теории Проектирования Йошикавы – Томиямы, и далее проследить в многочисленных модификациях теории FBS (функция – режим работы – структура) [5]. Эти теории предполагают взаимодействие и происхождение одного аспекта из другого, что с точки зрения последовательности фаз проектирования является абсолютно верным. Но для каждого аспекта вводится понятие

собственной модели знаний и, как следствие, необходимы переходы от одной модели к другой при каждом переходе к следующей фазе проектирования.

По этой причине большинство теорий считаются недостаточно формализованными, так как перечень моделей представления знаний определен жестко и не может быть дополнен либо изменен. На самом деле эти теории слишком формализованы, так как в них заранее формально описана каждая (но не все) фаза и модель, что не позволяет абстрагироваться от конкретных задач проектирования. Модели и требования не должны разделяться на какие-либо категории, в частности на функциональные и структурные.

Если определить некий формат описания требований, который удастся формализовать не в рамках специфики его описания, а универсально для всех возможных вариантов предъявления и описания требований, то разделение проектирования на фазы будет исключительно показательным, работа системы же будет осуществляться вне фаз проектирования.

Можно предположить, что основной причиной разделения процесса проектирования на фазы, стадии и этапы служит, видимо, защитная реакция человека на два основных фактора:

- а) недостаточный объем человеческой памяти, что требует участия разных специалистов для работы над одним достаточно сложным проектом;
- б) необходимость согласования частных («одноаспектных») решений, принятых различными разработчиками вне контакта между собой.

Таким образом, перенося в сферу автоматизированного проектирования все традиции организации проектирования неавтоматизированного, мы навязываем компьютеру свою несовершенную методику работы, порожденную ограниченностью не его (компьютерных), а наших (человеческих) возможностей.

В методиках проектирования большинства технических объектов последовательность этапов задана довольно жестко, например, для *самолета*: аэродинамика → динамика полета → конструкция и прочность → технология ...; для *электродвигателя*: электромагнитный расчет → тепловой расчет → вентиляционный расчет → расчет шумов и вибраций → механический расчет → расчет надежности и т.д.

Для автоматизированной системы оба указанных выше фактора не имеют решающего значения (при условии, что процедуры согласования решений также формализованы), она действительно могла бы работать «вне фаз проектирования» и для нее незачем столь строго устанавливать последовательность определения свойств различных групп. Но для этого нужны совершенно иные методики проектирования, не предполагающие разделения свойств ни на категории (функциональные, конструктивные, затратные и др.), ни по физической природе (механика, электродинамика, гидравлика и т.п.). Оценить вред от излишних ограничений пока трудно, однако очевидно, что все, что сковывает свободу проектировщика, будь то в содержании действий или в их последовательности, – не на пользу делу.

Таким образом, можно разработать максимально гибкую систему, работа в которой с различного рода требованиями будет единообразной. Делегировав расчеты и проверку соответствия требованиям по конкретным агентам, получим «проектировочный конструктор», пользоваться которым сможет любой специалист предметной области, накапливая при этом свои знания в базе знаний.

Каждый элемент этой базы знаний является агентом, у него есть собственные методы расчета и проверки соответствия требованиям. Поэтому при повторном проектировании работа проектировщика значительно упрощается

вследствие повторного использования спроектированных ранее элементов. При этом свойства элемента не делятся на структурные и функциональные. Свойства – это всего лишь параметры элемента, основной функцией которых является определение соответствия предъявленным требованиям, и только во вторую очередь они отвечают за визуализацию результата – физическую или функциональную модель проектируемого изделия.

С такой точки зрения различные виды проектирования: структурное, функциональное, концептуальное, перестают быть решающими факторами выбора стратегии проектирования, в том числе и автоматизированного. Многоаспектность не может быть достигнута до тех пор, пока количество аспектов определяется программистом, разрабатывающим программное обеспечение для специалистов предметных областей. Необходимо не расширять количество подходов (аспектов) к проектированию (такая тенденция просматривается в ряде работ: Общая Теория Проектирования Йошикавы–Томиямы – проектирование по требованиям структуры, затем расширение этой теории функциональным аспектом – Браха, спаренное проектирование [6]), а разработать один, универсальный подход, который определит N аспектов, где N будет зависеть непосредственно от конечного пользователя.

Проблема перехода от одной стадии проектирования к другой – это иллюзия, привнесенная в современное проектирование потребностью воссоздавать образ проектируемого изделия с каждой точки зрения заново, формируя новую модель для каждой грани проектного решения. Технолог и конструктор, работая над одной задачей, видят ее по-разному, но суть задачи от этого не меняется, и если бы знания технолога и конструктора были объединены, то проектирование двух моделей с последующим сопряжением и всеми вытекающими сложностями было бы бессмысленным.

Заметим, что с этой точки зрения узаконенный многими стандартами этап обеспечения технологичности конструкции изделия (ТКИ) между конструированием и технологической подготовкой производства – один из наиболее одиозных результатов нашего искусственного перехода к последовательному проектированию: его основное содержание как раз и состоит в пересмотре части конструкторских решений, принятых без учета мнений технологов (т.е. в рамках одного аспекта). Здесь мы видим всеми громогласно осуждаемое «повторное проектирование», но возведенное во всем мире в ранг обязательной процедуры.

Преимущество автоматизированной системы заключается в том, что она может содержать в себе неограниченный объем знаний различных предметных областей, оперируя при этом одной моделью. В таком случае не нужно, как это утверждается в работах Брахи, проверять сначала соответствие структурным требованиям, а затем искать пересечения с удовлетворением функциональным требованиям, нужно просто обеспечить соответствие всем требованиям. Тем более, что при решении той или иной задачи все же возникнет потребность в дополнительных аспектах.

По большому счету функциональное проектирование – это всего лишь определение перечня функциональных свойств (требований), которым должен соответствовать проектируемый объект. И даже если удастся найти такое решение функциональной головоломки, которое будет полностью удовлетворять поставленным требованиям, то нет гарантии, что соответствующая выбранным функциональным решениям структура удовлетворит структурным требованиям.

Поэтому необходимо рассматривать и структурные, и функциональные, и другие требования как единую сущность.

Следовательно, система проектирования должна состоять из агентов, обладающих рядом свойств, деление которых на какие-либо категории весьма условно и введено исключительно для удобства оперирования конечным пользователем. Каждый агент располагает расчетными методами и методами проверки соответствия некоторому требованию. Кроме того система располагает рядом требований, каждый агент может соответствовать или не соответствовать какому-либо требованию. При этом параметрические и структурные ограничения также являются требованиями.

В такой системе проектирование сводится к выбору набора агентов, удовлетворяющих всем поставленным требованиям. Задача расчета параметров и структуры является вложенной задачей выбора, так как при определении выполнения или невыполнения требования происходит расчет всех возможных реализаций агента.

Последовательность фаз проектирования неявно учитывается при расчете модели, так как, чтобы определить значение свойства, как правило, используют ранее определенные свойства. А в общем случае изначально определяются свойства аспектов более ранних фаз проектирования, например от функции к структуре.

В соответствии с нашими требованиями агент является в высокой степени мобильной сущностью, которая может остановить свое выполнение, изменить внутреннюю структуру и продолжить выполнение.

Укрупненная физическая структура агента показана на рис. 3. Дадим определение каждой сущности, которая изображена на рисунке:

- *Агент.* Определим агента как некую автономную сущность, которая способна ощущать и воздействовать посредством рецепторов и эффекторов. В нашем случае программной реализации восприятие и воздействие будет осуществляться путем передачи сообщений. Многие при определении агента акцентируют внимание на его интеллектуальности [7]. Свойство интеллектуальности зависит от конкретной реализации поведения агента и напрямую не зависит от его архитектуры. К архитектуре в этом случае предъявляется только несколько минимальных требований – она должна быть по крайней мере такая, что позволяет проявлять интеллектуальность. В нашем случае это можно осуществить через каналы ввода/вывода (ощущение/воздействие). Механизмы наделения агентов интеллектуальностью выходят за рамки темы этой статьи.

- *Внешняя среда.* Внешняя среда определяется как совокупность объектов, не принадлежащих данному агенту и воспринимаемых им как отдельная сущность. Оттуда поступают сигналы, которые агент может проигнорировать, если он не знает рода сигналов, либо ответить каким-либо действием. Таким образом, если определить внешнюю среду как множество агентов, то для агента без рецепторов и эффекторов данное множество будет пустым. Чем больше множество, тем больше агент знает об окружающей среде.

В нашем случае любой агент должен обладать базовой функциональностью, определенной в модуле управления агентом, поэтому каждый агент знает о существовании других агентов в зоне досягаемости. Можно сравнить базовую функциональность агента с интерфейсом IUnknown архитектуры COM, в дальнейшем можно провести некоторую аналогию между методами, определяемыми этим интерфейсом.

- *Модуль управления агентом* – данный модуль является статической частью агента и отвечает за реализацию спецификации агента в памяти, передачи ему входящих сообщений, событий, за отправку сообщений во внешнюю среду. Сразу необходимо оговорить, что на этом уровне будет происходить синхронизация с другими агентами, блокировки доступа к агенту, разрешение агентных ссылок, остановка и запуск агента, сохранение его состояния.

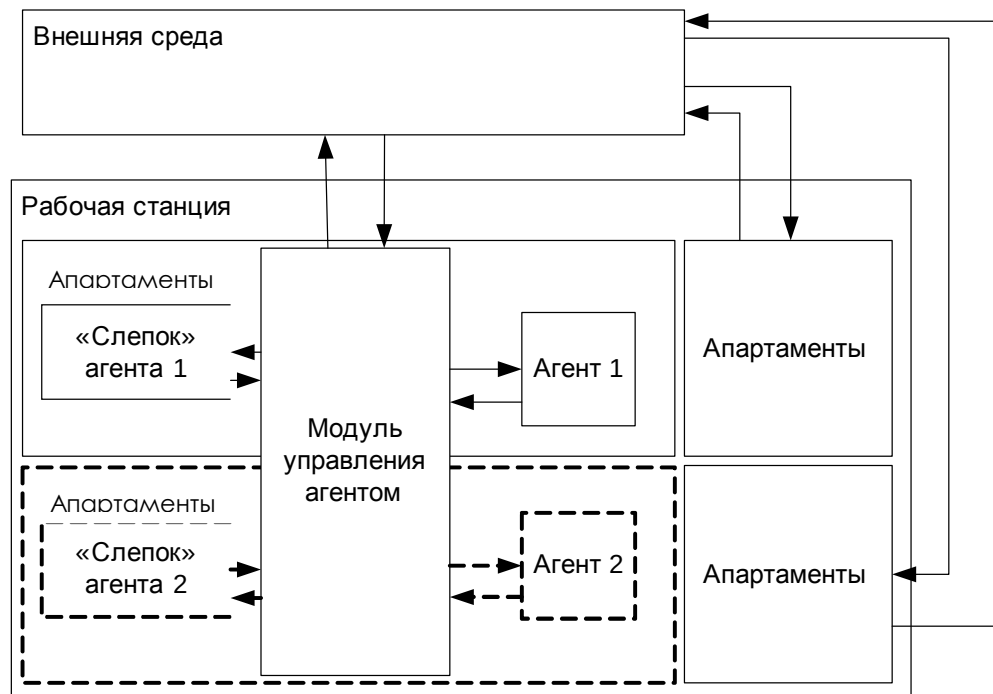


Рис. 3. Укрупненная структура агента

- *«Слепок» агента* – таким образом мы обозначим все мета- и динамические данные об объекте: его спецификацию, последнее сохраненное состояние, ассоциированные данные.

- *Апартаменты агента* – это тот набор данных и работающих модулей, которые относятся к конкретному агенту.

Необходимо пояснить суть отношений между модулем управления агентом и самим агентом. На диаграмме представлен случай, когда один модуль управления может управлять несколькими агентами. Здесь можно провести аналогию с сервером CORBA, который при получении нового запроса передает его адресату – конкретному экземпляру объекта. В нашем случае ситуация несколько иная, но суть остается неизменной: практически модуль управления агентом – это его серверная часть, которая передает в домен агента предназначенные ему сообщения. Создание нового агента может происходить как с клонированием мобильной части, например, если мы хотим новый агент переместить в другую среду выполнения, грубо говоря, на другой компьютер, либо без клонирования мобильной части, тогда одному модулю управления будет соответствовать несколько агентов. Однако в дальнейшем будем принимать модуль управления и агента как одно целое. Процесс клонирования модуля управления должен происходить прозрачно для конечного создателя агента.

Приняв такую архитектуру, получим мобильного агента, для существования которого не нужно дополнительных компонент.

Рассмотрим некоторую задачу, абстрагируясь от проектирования и акцентируя внимание на решении задачи выбора.

Допустим, нам надо купить собаку в зоомагазине. Собака должна быть взрослой, рост собаки не должен превышать 50 см в холке, она должна уметь приносить тапки и любить хозяев. Купить собаку мы можем только, если приобретем для нее намордник.

Переведем задачу на язык агентов.

В агентной сети «Зоомагазин» необходимо выбрать агент типа Собака, удовлетворяющий следующим требованиям: возраст более одного года; рост в холке – менее 50 см; умеет приносить тапки; любит хозяев; есть подходящий намордник.

Необходимо также выбрать агент типа Намордник, удовлетворяющий условию: размер должен быть равен N, где N – размер агента типа Собака.

В сети «Зоомагазин» есть три агента типа Собака. Соответствие их предъявленным требованиям представлено в табл. 2.

Таблица 2

Соответствие агентов предъявленным требованиям

Собака	Сенбернар по кличке Тузик	Ротвейлер по кличке Шарик	Такса по кличке Бобик
Возраст	2 года	6 месяцев	1 год
Рост в холке	45 см	30 см	25 см
Умение приносить тапки	+	+	+
Любит хозяев	+	-	+
Размер намордника	5	4	2

Требованиям соответствуют два варианта агента типа Собака. Для определения соответствия по пятому требованию выберем агента типа Намордник.

В сети есть агенты Намордник1 и Намордник2. Размер агента должен соответствовать одному из размеров агента типа Собака, которые удовлетворяют условиям выбора агента типа Собака. Намордник1 имеет размер 5, Намордник2 имеет размер 4. Таким образом, предъявленным требованиям удовлетворяет агент Намордник1.

Проверяем на соответствие требованию 5 агентов типа Собака. Теперь мы знаем, что размер агента типа Намордник равен 5, следовательно, условию соответствует только агент Сенбернар по кличке Тузик.

Решение поставленной задачи: выбор агента типа Собака – Сенбернар по кличке Тузик, выбор агента типа Намордник – Намордник1. Или, выражаясь терминами предметной области – купить сенбернара, которого зовут Тузик, и намордник пятого размера.

Следует отметить, что при решении задачи мы не разделяли требований на какие-либо категории и не выясняли, к какой категории отнести возраст собаки или ее любовь к хозяевам. Выбор осуществлялся одновременно по всем критериям. Мы не выбирали сначала собак, соответствующих структурным требованиям (рост в холке), и не пытались затем привести эти требования к

функциональной модели (существо, умеющее приносить тапки). Такие задачи решаем, не задумываясь над декомпозицией требований на категории, поскольку наша база знаний позволяет представить собаку целостно и с точки зрения структуры, и с точки зрения функциональности.

Очевидно, что, усложнив задачу и заменив выбор собаки проектированием самолета, даже самый опытный специалист не сможет справиться с представлением целостности. Именно здесь истоки мифа о неизбежности разделения процесса проектирования на аспекты с различными моделями. Но автоматизированная система может располагать базой знаний неограниченной сложности и решение задачи выбора собаки и выбора двигателя для самолета будет проводиться по одному и тому же алгоритму.

Выбор типа двигателя может быть выполнен на основании зависимости тягового КПД от числа M полета [8] (рис.4).

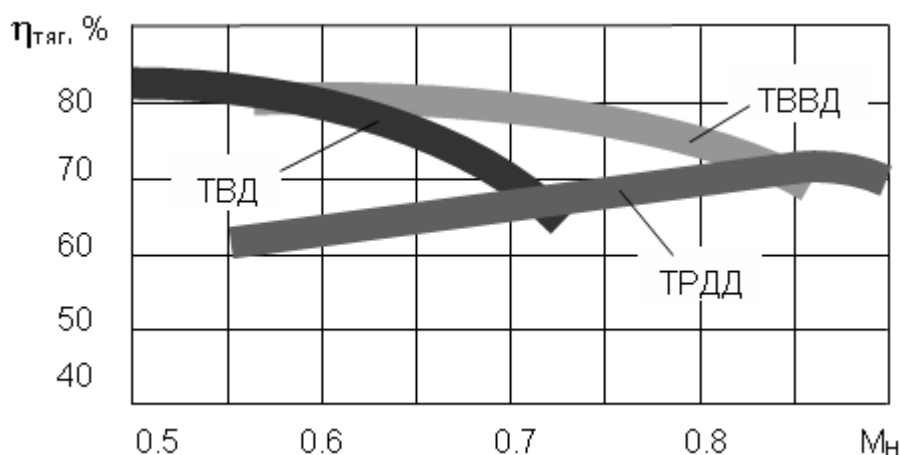


Рис. 4. Зависимость тягового КПД двигателя от числа M полета

Механизмом соответствующего модуля базы знаний служит таблица решений (табл. 3). При этом не следует понимать выбор агента как выбор готового результата из уже существующего перечня. Расчет параметров и определение структуры в проектировании – это тоже своего рода выбор значений или элементов изделия, как, например, расчет размеров двигателя в табл. 3. В случае описанной выше многоаспектной агентной сети (базы знаний) к этим параметрам выбора добавляется еще выбор метода расчета, т.е. выбор решения в виде того или иного агента.

Таблица 3

Таблица решений для определения типа и характеристик двигателя

Число M полета	До 0,55	0,55..0,83	Свыше 0,83
Тип двигателя	ТВД	ТВВД	ТРДД
...	...	...	...
Диаметр двигателя (винта), м	$(N_0/8,1)^{0.25}$	...	$0,1 \cdot G_B^{1.5} \cdot \varphi_1(R_0) \cdot \varphi_2(\pi_k^*) \cdot \varphi_3(T_3)$
Примечание: N_0 – мощность; G_B – секундный расход воздуха; R_0 – стартовая тяга; π_k^* – степень повышения давления в компрессоре; T_3 – температура газа перед турбиной; φ_i – статистические коэффициенты			

Для автоматизированной системы не имеет значения и традиционное деление тактико-технических требований (ТТТ) на тактические и технические: из тактических (в общеинженерном понимании «функциональных») требований за одну итерацию извлекаются не только технические требования, но и ряд конструктивных («структурных») признаков, например с использованием продукционных правил вида:

ЕСЛИ ТТТ предусматривают обычные условия применения самолета (например, дозвуковой пассажирский или транспортный),

ТО структура самолета должна включать в себя: фюзеляж, крыло, оперение, шасси.

ЕСЛИ заданная крейсерская скорость существенно превышает посадочную (определяемую классом аэродрома, заданным в ТТТ),

ТО в структуру крыла следует включить закрылок вместе с узлами его навески.

Отсюда следует, что в условиях компьютерной поддержки принятия проектных решений нет необходимости задерживать начало работы конструкторов до полного формирования облика изделия, а технологов – до оформления рабочей документации. И задача современных разработчиков – преодолеть барьер формализации знаний, представить их как некоторую сущность, соответствующую единому, гибкому и масштабируемому шаблону.

Список литературы

1. Егер С.М. Основы автоматизированного проектирования самолетов / С.М. Егер, Н.К. Лисейцев, О.С. Самойлович. – М.: Машиностроение, 1986. – 232 с.
2. Лазарев И.А. Композиционное проектирование сложных агрегативных систем / И.А. Лазарев. – М.: Радио и связь, 1986. – 326 с.
3. Гайнутдинова Т.Ю. Программный комплекс автоматизированного проектирования авиационных конструкций / Т.Ю. Гайнутдинова // Вестник ТГГПУ. Авиационная техника. – 2007. – № 2. – С. 67-70.
4. Cooper Dave, van Tooren Michel, Wan Mazlina Wan Mohamed. Keys to Success with Knowledge-based Techniques // SAE Paper Number 2008-01-2262.
5. Tomiyama T., Yoshikawa H. Extended General Design Theory // Design theory for CAD, Proceedings from IFIP WG 5.2, 1987, Elsevier Science Publishers B.V., Amsterdam. – P. 95-130.
6. Braha D., Reich Y. Topological structures for modeling engineering design processes // Eng Design 14 (2003). – P.185–199.
7. Веб-ресурс компании The Object Management Group (OMG) // <http://www.omg.org/>.
8. Нечаев Ю.Н. Авиационные турбореактивные двигатели с изменяемым рабочим процессом для многорежимных самолетов / Ю.Н. Нечаев, В.Н. Кобельков, А.С. Полев. – М.: Машиностроение, 1988. – 176 с.

Рецензент: д-р техн. наук, проф. В.Г. Сухоребрый, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

Поступила в редакцию 23.03.2010

Архітектура багатоаспектної платформи для вирішення задач структурно-параметричного синтезу об'єктів. Багатоаспектне моделювання

Показано необхідність доповнення агентної архітектури розподілених САПР багатоаспектною моделлю об'єкта, що дозволяє підвищити гнучкість подання інженерних знань та управління процесом проектування. Обґрунтовано можливість переходу в рамках автоматизованого проектування від жорсткої послідовності стадій та етапів до паралельного виконання проектних процедур.

Ключові слова: багатоаспектна модель, структурний синтез, параметричний синтез, база знань, програмний агент, агентна мережа.

Multifaceted platform architecture for structure and parametric synthesis tasks solution. Multifaceted modelling

The necessity of allocated CAD agent architecture addition with the multifaceted model of development object is shown. This addition allows to promote engineering knowledges and design process control representation flexibility. Transition possibility is grounded from hard stages subsequence to parallel execution of project procedures at the automated design.

Keywords: multifaceted model, structural synthesis, parametric synthesis, knowledge base, software agent, agent network.