

УДК 658.512.22 + 004.925.8

Д.Э. Цыганков, А.Ф. Похилько

ПРЕДСТАВЛЕНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ НА БАЗЕ ОБОБЩЕНИЯ ЭЛЕМЕНТАРНЫХ ОПЕРАЦИЙ ДО УРОВНЯ СЕМАНТИЧЕСКИХ ЕДИНИЦ

Цыганков Денис Эдуардович, окончил радиотехнический факультет Ульяновского государственного технического университета, аспирант кафедры «Прикладная математика и информатика» УлГТУ. Имеет статьи в области системного анализа и автоматизации процессов проектной деятельности. [e-mail: d.tsyg@mail.ru].

Похилько Александр Федорович, кандидат технических наук, доцент, окончил факультет технической кибернетики Ленинградского политехнического института им. М.И. Калинина. Профессор кафедры «Прикладная математика и информатика» УлГТУ. Имеет статьи в области принятия решений и интегрированных процессов в системах автоматизированного проектирования. [e-mail: afp@ulstu.ru].

Аннотация

Представление процесса проектирования системой проектных процедур, описываемых в рамках методологии IDEFx, позволяет декомпозировать его на проектные стадии – последовательности действий формирования проектного решения, состоящие из проектных операций или процедур и имеющие законченность с точки зрения получения имеющего реальное смысловое содержание объекта или его фрагмента (физический смысл). Каждая из таких стадий строго определена как в плане порядка ее выполнения, так и в плане исходных и выходных данных, что делает невозможным внесение в нее изменений с сохранением корректности общей структуры формирования проектного решения. В настоящей работе исследуется модель представления процессов проектной деятельности с использованием вводимого понятия проектной стадии как семантической единицы и с использованием технологии Open CASCADE. В рамках исследования авторами выделены множества проектных условий и ограничений на выполнение проектных операций, на основании которых сформирована система проектных процедур из состава библиотек Open CASCADE Technology, программная реализация которых способна обеспечить модифицируемость проектных стадий. Таким образом, предлагаемое представление процесса проектирования позволяет накапливать, обобщать и модифицировать процесс формирования проектных решений, при этом сохраняя целостность как процесса, так и проектного решения.

Ключевые слова: автоматизация, проектная стадия, процесс, проектная деятельность, модель, семантика, проектная процедура, Open CASCADE Technology, решение, 3D-образ.

REPRESENTING DESIGN PROCESS BY GENERALIZING ELEMENTARY OPERATIONS TO SEMANTIC UNIT LEVEL

Denis Eduardovich Tsygankov, Post-Graduate Student at the Department of Applied Mathematics and Informatics of Ulyanovsk State Technical University; graduated from the Faculty of Radioengineering at Ulyanovsk State Technical University; an author of articles in the field of system analysis and design process automation. e-mail: d.tsyg@mail.ru.

Aleksandr Fedorovich Pokhilko, Candidate of Engineering, Associate Professor; graduated from Leningrad Polytechnic Institute named after M.I. Kalinin; Professor at the Department of Applied Mathematics and Informatics at Ulyanovsk State Technical University; an author of articles in the field of decision-making and integrated processes of CAD-systems. e-mail: afp@ulstu.ru.

Abstract

The design process can be represented by the system of project procedures described in the IDEFx methodology. That allows to decompose it into design stages. Design stage is a sequence of design solution formation actions. Such sequence consists of project operations or procedures and has completeness in terms of getting an object or its fragment with real semantic content (physical meaning). Each stage is strictly defined in terms of either its execution order or the input and output data that makes it impossible to change them and preserve the overall structure correctness of the design solution formation. The authors research the model of project activity processes representation with the use of the introduced concept of the design stage as a semantic unit and Open CASCADE Technology. The researches examine a lot of design conditions and limitations. On the basis of such conditions and limitations, with the use of Open CASCADE Libraries, the system of project procedures was developed. The software implementation of these procedures can provide the modifiability of project stages. Therefore,

the representation of design process allows to accumulate, generalize and modify the process of design solution formation preserving the completeness of both the process and design solution.

Key words: automation, design stage, process, design activity, model, semantics, design procedure, Open CASCADE Technology, solution, 3D-image.

ВВЕДЕНИЕ

Инфокоммуникационные технологии в настоящее время полностью определяют стратегию развития производственных систем, обеспечивающих формирование и функционирование производственных структур, взаимодействие между подразделениями которых осуществляется на базе современных средств телекоммуникаций с использованием CALS-технологий на всех стадиях жизненного цикла изделий (проектирование, производство) [1]. Развитие соответствующих инфокоммуникационных технологий выводит на передний план необходимость исследования и развития методов представления проектной информации, обеспечивающих не столько обмен между различными информационными системами или сервисами, но в первую очередь поддержку хранения, модификации и обобщения информационных представлений проектных решений [2]. В настоящей работе исследуется возможность семантического представления проектных стадий, не только не нарушающего целостность формирования проектных решений, но и позволяющего обобщить их в форме описания класса проектных решений.

ПРОЦЕДУРНОЕ ПРЕДСТАВЛЕНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ

Процесс проектирования в общем случае может быть представлен последовательностью проектных процедур [3], каждая из которых обладает определенным физическим смыслом. Простейший вариант – линейное выполнение проектных процедур – представлен на рисунке 1 слева.

На вход первой процедуры поступает техническое задание (ТЗ) Rs_{in} , представляющее собой совокупность исходных данных и требований к проектируемому изделию. Каждая следующая процедура выполняется после успешно выполненной предыдущей, а входными данными к i -й процедуре являются выходные данные $(i-1)$ -й процедуры [4]. При этом внутри самих процедур возможны различные варианты преобразования исходных данных в выходные ветви-альтернативы.

Для линейной последовательности процедур проектное решение Ds_{out} формируется по следующей формуле:

$$Ds_{out} = (Do_1, 1) \cup (Do_2, 1) \cup \dots \cup (Do_n, 1). \tag{1}$$

В такой последовательности могут быть получены различные проектные решения, описываемые формулой (1). Однако данная

последовательность способна описать лишь класс геометрически подобных объектов, так как не обеспечивает вариативность выполнения проектных процедур. Последовательность, имеющая на 1-й проектной стадии два варианта проектных маршрутов, представлена на рисунке 1 в центре.

Значения входных и промежуточных проектных параметров определяют последующую проектную процедуру на данной стадии – 1Do_2 или 2Do_2 , каждая из которых в результате приведет к оригинальному проектному решению ${}^iDs_{out}$.

Для данной последовательности решение Ds_{out} выражается формулой:

$$Ds_{out} = (Do_1, 0) \cup \{[(Do_2, 1) \cup \dots \cup (Do_n, 1)] \cap [(Do_2, 2) \cup \dots \cup (Do_n, 2)]\}. \tag{2}$$

Типичным примером может быть построение 3D-модели цилиндрической детали: она может быть получена как разными программными средствами, так и разными путями (путем вытягивания эскиза, его вращением, построением по сечениям и др.).

Наиболее общий случай – древовидная последовательность проектных процедур, каждая из которых имеет несколько ветвей альтернатив, представлен на рисунке 1 справа.

В общем случае в результате выполнения процедуры на i -й проектной стадии на $(i+1)$ -й стадии образуется n возможных процедур. В результате чего данная последовательность позволяет получать на выходе множество проектных решений, каждое из которых является уни-

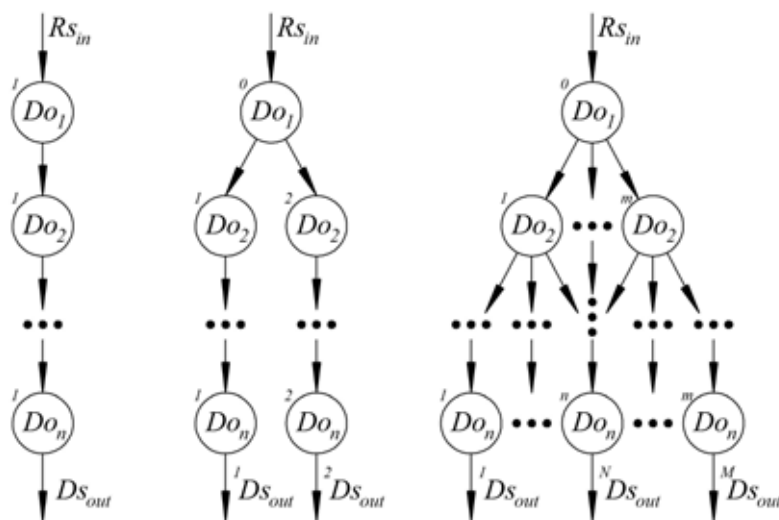


Рис. 1. Варианты последовательностей проектных процедур

кальным, но при этом учитывает специфику дерева проектных маршрутов (как в случае проектирования объектов, входящих в единый класс).

Для общего случая последовательности проектных процедур проектное решение Ds_{out} формируется по формуле:

$$Ds_{out} = (Do_1, 0) \cup \left\{ \left[(Do_2, 1) \cup ((Do_3, 1.1) \cap \dots \cap (Do_n, 1.n)) \right] \cap \right. \\ \left. \cap \left[(Do_2, 2) \cup ((Do_3, 2.1) \cap \dots \cap (Do_n, 2.n)) \right] \cap \dots \cap \right. \\ \left. \cap \left[(Do_2, m) \cup ((Do_3, m.1) \cap \dots \cap (Do_n, m.n)) \right] \right\}. \quad (3)$$

Представленная последовательность может быть использована при проектировании сложных технических объектов, так как каждая ветвь проектного маршрута может характеризоваться определенным свойством, отличающим одно решение от других.

В качестве примера изделия рассматривается нагрузка согласованная радиочастотная коаксиальная типа III (вилка) по ГОСТ РВ 51914-2002, сборочная 3D-модель которой представлена на рисунке 2.



Рис. 2. Сборочная 3D-модель нагрузки согласованной коаксиальной

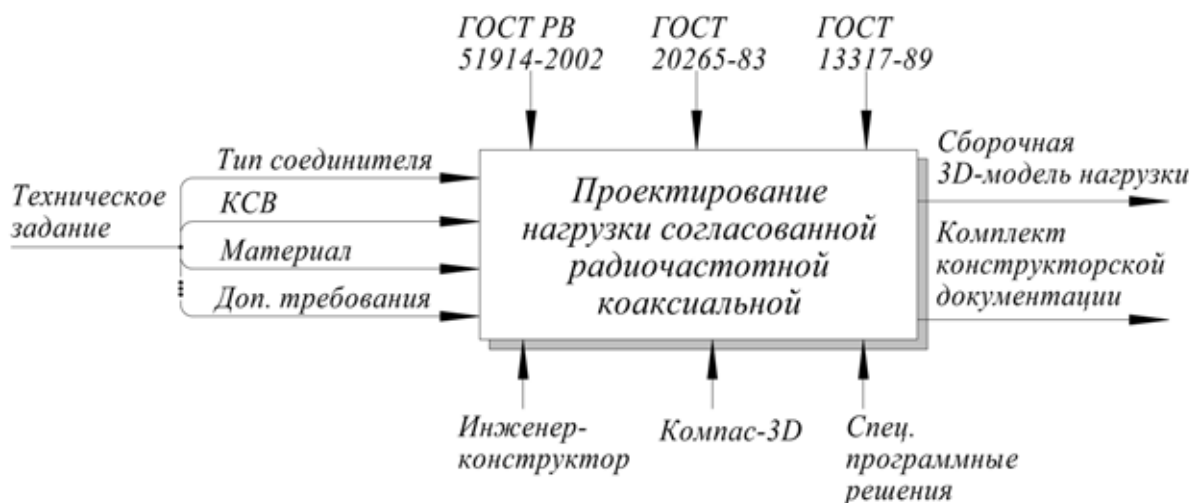


Рис. 3. Контекстная IDEF0-модель процесса проектирования нагрузки коаксиальной

В рамках методологии IDEFx проектирование нагрузки коаксиальной есть процесс [5], исходными данными которого являются ТЗ и дополнительные технические требования. ТЗ может включать в себя такую входную информацию, как тип соединителя по ГОСТ 20265-83, значения коэффициента стоячей волны (КСВ), требования по материалу, габаритные размеры и т.д. [6]. Данный процесс выполняется инженером-конструктором при помощи различных систем автоматизированного проектирования (САПР): электромагнитного, твердотельного и др. моделирования, а также оформления документации; ввод управляющих и исходных данных задается посредством используемых стандартов (ГОСТ 13317-89, ГОСТ РВ 51914-2002 и др.) и нормативно-технической документации (ВРО.364.039ТУ). В результате проектирования на выходе формируется решение в виде выпуска комплекта конструкторской документации и информационного 3D-образа.

На рисунке 3 представлена контекстная диаграмма процесса проектирования нагрузки согласованной коаксиальной, представленная на основе методологии IDEF0.

Декомпозиция контекстной диаграммы представляет собой упорядоченную последовательность взаимосвязанных проектных процедур с указанием входных и выходных данных, а также управляющих воздействий на каждую из них.

В данном случае контекстная диаграмма будет включать в себя такие процедуры, как «Проектирование стержня», «Проектирование втулки», «Формирование сборки» и др. При этом важно, что ТЗ определяет параметры каждой из деталей, входящих в изделие, следовательно, изначально задав диаметр стержня $d=1,7$ и тип вилки III по ГОСТ 20265-83 [7], уже будет невозможно изменить параметры прочих деталей, определяемых этими стандартами, что в результате обеспечит корректную сборку изделия.

Данные, определяющие итоговую сборку, могут быть заданы не только в ТЗ, но и в рамках выполнения *i*-й проектной процедуры. Далее рассматривается присоединительная гайка (рис. 4), входящая в состав соединителя и служащая для фиксации двух коаксиальных радиочастотных приборов.

Такие параметры, как количество и глубина боковых вырезов (под ключ), тип и длина рифления, могут не задаваться в ТЗ, а определяться лишь на уровне данной процедуры. При этом, в случае принятия различных локальных проектных параметров, будут получены различные детали, а следовательно, и изделия будут различны, так как они будут соответствовать различным проектным маршрутам.



Рис. 4. 3D-модель гайки присоединительной и IDEF0-модель процесса ее проектирования

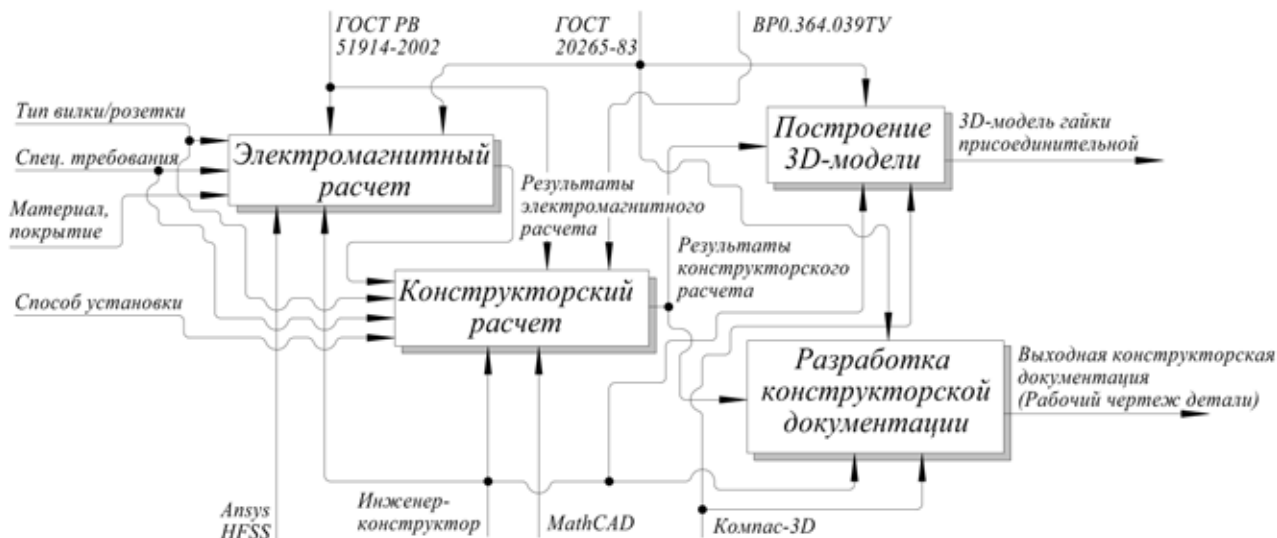


Рис. 5. Декомпозиция процесса проектирования гайки присоединительной

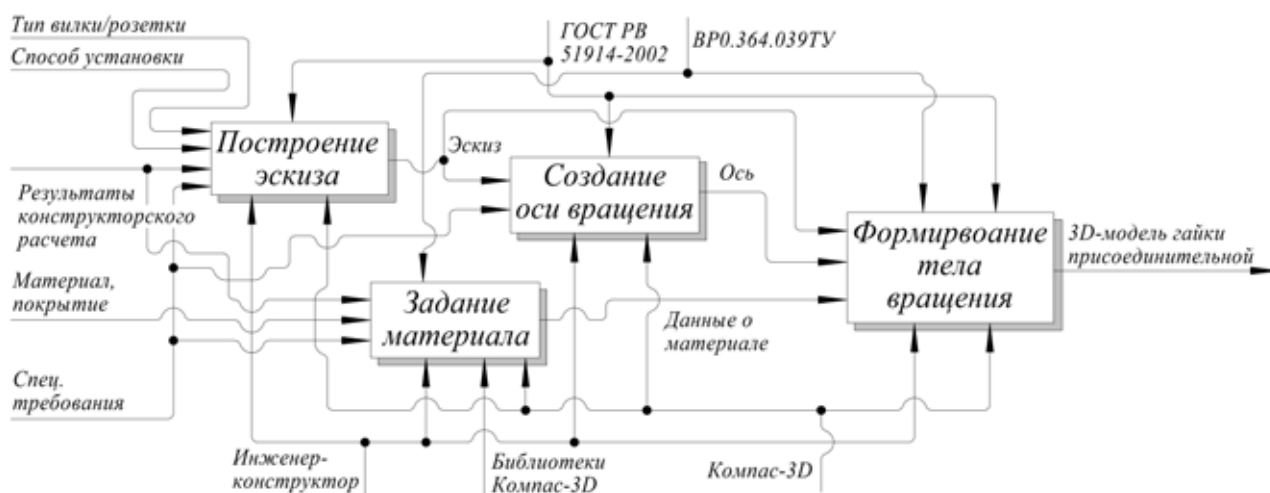


Рис. 6. Декомпозиция процедуры построения 3D-модели гайки присоединительной

На рисунке 4 также представлена IDEF0-модель процесса проектирования гайки присоединительной.

Данные для проектирования детали – это лишь часть исходного ТЗ, оказывающая непосредственное влияние на формируемое проектное решение. Процессная диаграмма, представленная на рисунке 4, декомпозируется на такие процедуры, как электромагнитный и конструкторский расчеты, оформление конструкторской документации и формирование 3D-модели (рис. 5).

Подробно рассматривается проектная процедура построения трехмерной модели детали. Средствами практически любых современных САПР данная 3D-модель (рис. 4.) наиболее легко может быть построена как тело вращения. В таком случае проектная процедура построения 3D-модели гайки декомпозируется следующим образом (рис. 6).

СТРУКТУРНЫЙ АНАЛИЗ ПРОЦЕССА ПОСТРОЕНИЯ 3D-МОДЕЛИ

Последовательность проектных операций, представленная на рисунке 6, является идентичной практически для всех современных САПР. В общем случае данная последовательность основана на следующих правилах:

1. Чтобы получить тело, нужно выполнить некоторые действия с замкнутым контуром.
2. Чтобы сформировать замкнутый контур, необходимо создать эскиз.
3. Чтобы создать эскиз, нужно выбрать базовую плоскость.

Структура построения твердого тела, представленная на рисунке 7, демонстрирует, что главными элементами тела являются эскиз и операция над ним. При этом важно, что алгоритмы, заложенные в САПР, проверяют возможность выполнения i -й процедуры и допускают ее лишь при корректных условиях. Например, невозможно вытянуть или вращать незамкнутый контур, построить первым элементом вырез, построить тело по сечениям по одному эскизу и др. Данные алгоритмы и ограничения закладываются разработчиками, а так как такие современные САПР, как CATIA, SolidWorks и КОМПАС-3D, являются универсальными системами 3D-моделирования, то эти алгоритмы должны обеспечивать определенную свободу действий пользователя.

Алгоритмы, заложенные в современные САПР, автоматизируют сложные и не нуждающиеся в ручной настройке операции (преобразование грани в замкнутый контур), оставляя только функционал, понятный пользователю.

Исследуемая открытая платформа Open CASCADE Technology включает в свой состав наборы библиотек и средств разработки систем 3D-моделирования [8]. Главными преимуществами данной платформы являются возможность самостоятельно задавать алгоритмы, условия и ограничения на выполнение операций и практически полная параметризованность [9, 10]. Структура построения твердого тела средствами Open CASCADE Technology имеет следующую последовательность (рис. 8.)

Как видно, для построения твердого тела требуется выполнение намного большего числа операций (с большим числом проектных параметров), но при этом становится возможным на уровне разработчика выделить исходные данные, связав их с последовательностью операций, а также ограничить функционал, оставив лишь функции, необходимые для выполнения проектной процедуры, и тем самым программно реализовать инструмент выполнения конкретной проектной процедуры [11].



Рис. 7. Структура построения твердого тела современными САПР

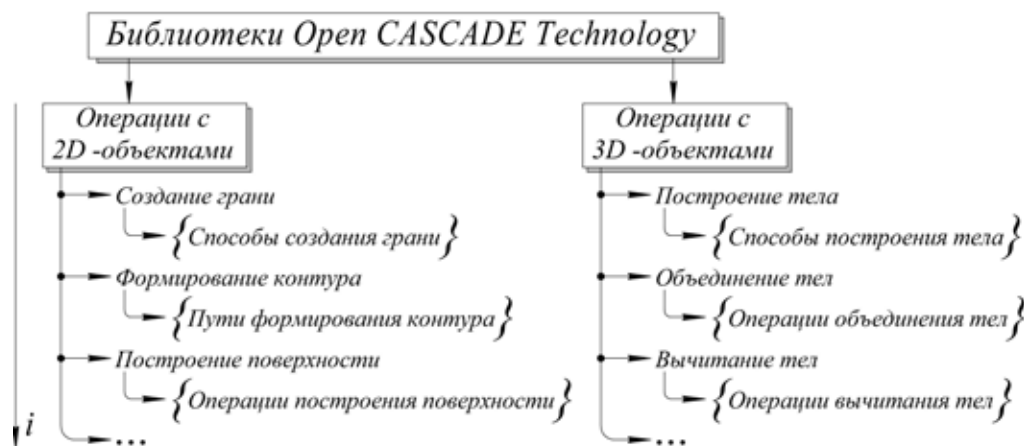


Рис. 8. Структура построения твердого тела на основе Open CASCADE Technology

Таким образом, если для построения 3D-модели гайки необходимо выполнение последовательности, представленной на рисунке 6, то подход на основе технологии Open CASCADE будет включать в свой состав те же операции, только они будут выполняться не инженером-конструктором вручную, а автоматизировано, в зависимости от введенных исходных данных. Основной сложностью данного метода является программная реализация всех алгоритмов, ограничений и условий, применяемых к проектируемой детали. В случае с присоединительной гайкой исходными данными для проектирования будут не только те, что представлены на рисунке 4 (определяемые ТЗ и влияющие на изделие), но и локальные данные, влияющие на детали (например, тип рифления, маркировка, угол фаски и др.), которые также определяют итоговый вид проектируемого изделия.

СЕМАНТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СТАДИЙ ПОСТРОЕНИЯ 3D-МОДЕЛИ

Семантическое представление процесса построения 3D-модели изделия $DesPr_1$ (рис. 5), основанное на программном кодировании проектных процедур с использованием библиотек Open CASCADE Technology $DesPr^{OCC}$, возможно лишь при предварительном определении следующих трех множеств:

1) Множество проектных условий выполнения проектных процедур $MCon^{Proc.imp.}$:

$$MCon^{Proc.imp.} = \left\{ dCon_1^{Proc.imp.1}, dCon_2^{Proc.imp.1}, \dots, dCon_i^{Proc.imp.k} \right\}, \quad (4)$$

где $dCon_i^{Proc.imp.k}$ – это i -е условие выполнения k -й проектной процедуры $Proc_k$ на проектной стадии St_i .

Как правило, в процессе выполнения процедуры на каждой стадии St_i требуется ряд проектных условий. Например, для построения тела вращения должны выполняться не только стандартные условия замкнутости и отсутствия самопересечения контура, но и наличие осевой линии, относительно которой будет вращаться контур.

Определение множества $MCon^{Proc.imp.}$ задает структуру формирования решения, которая будет корректной, но не единственно возможной для каждого множества условий. Следовательно, выделение всей совокупности проектных стадий $MdProc$ и дальнейшее определение системы накладываемых на них условий зададут строгую последовательность проектных маршрутов.

2) Множество допустимых проектных процедур, выполнение которых потенциально возможно на данной проектной стадии $MPos^{Proc.st.}$:

$$MPos^{Proc.st.} = \left\{ dPos_1^{Proc.st.1}, dPos_2^{Proc.st.1}, \dots, dPos_i^{Proc.st.k} \right\}, \quad (5)$$

где $dPos_i^{Proc.st.k}$ – это i -я проектная процедура, которая может быть выполнена на i -й стадии St_i процесса проектирования.

На каждой проектной стадии St_i допустимо выполнение N проектных процедур: от единственно возможной (построение оси) до целого множества (построение тела из контура).

Допустимые процедуры могут быть связаны двумя способами: либо проектная процедура $dPos_1^{Proi.st.1}$ является взаимоисключающей относительно других операций стадии St_i (выбор способа построения тела из эскиза), либо дополняющей другие операции (последовательное задание входных данных при построении отрезка).

3) Множество ограничений на значения проектных параметров $MPar^{Lim.}$:

$$MPar^{Lim.} = \left\{ mPar_1^{Rang}, mPar_2^{Disc}, mPar_3^{Type} \right\}. \quad (6)$$

Данное множество состоит из трех подмножеств:

1. Подмножество диапазонов значений проектных параметров $mPar_1^{Rang}$, которое представляет собой заданные максимальное и минимальное значения i -го параметра. Численная мера различия двух ближайших значений, удовлетворяющих нахождению в интервале $(Par_{min}^{Rang} \dots Par_{max}^{Rang})$, определяется шагом дискретизации Δt . Значение проектного параметра должно удовлетворять условию принадлежности к данному диапазону значений:

$$Value(dPar_i) \in \left\{ mPar_1^{Rang} \mid \left(Par_{min}^{Rang} \dots Par_{max}^{Rang} \right) \right\}, \quad (7)$$

где $Value(dPar_i)$ – значение i -го проектного параметра.

2. Подмножество нормализованных значений проектных параметров $mPar_2^{Disc}$, представляющее собой дискретный список допустимых значений. Значение проектного параметра должно удовлетворять условию соответствия нормализованным значениям:

$$Value(dPar_i) \in \left\{ mPar_2^{Disc} \mid \left[vPar_1^{Disc}, vPar_2^{Disc}, \dots, vPar_n^{Disc} \right] \right\}, \quad (8)$$

где $vPar_i^{Disc}$ – i -е нормализованное значение параметра $dPar_i$.

3. Подмножество типов проектных параметров $mPar_3^{Type}$, представляющее собой перечень допустимых типов проектных параметров (integer, float, double). Тип проектного параметра должен удовлетворять условию соответствия предварительно заданным типам:

$$Type(dPar_i) \in \left\{ mPar_3^{Type} \mid \left[tPar_1^{Type}, tPar_2^{Type}, \dots, tPar_n^{Type} \right] \right\}, \quad (9)$$

где $Type(dPar_i)$ – тип значения i -го проектного параметра, $tPar_i^{Type}$ – i -й тип значения параметра $dPar_i$.

Проектные параметры, являясь аргументами проектных операций, определяют как сам результат, так и способ его достижения [12]. Проектные параметры на уровне операций и процедур определяют результат их выполнения, а на уровне стадии – решение и проектный маршрут.

При определении вышеперечисленных множеств наступает очередь их программной реализации (кодирования проектных процедур и выстраивания их в требуемом для конкретного проектного решения порядке), определяемая следующим выражением:

$$MDes^{OCC} = \sum_{i=0}^n \left(MLib^{OCC} \cap \left(MCon_i^{Proc.imp.} \cup MPos_i^{Proc.st.} \cup MPar_i^{Lim.} \right) \right), \quad (10)$$

где $MDes^{OCC}$ – множество проектных процедур, основанных на технологии Open CASCADE, ограниченных по функционалу для формирования 3D-модели конкретного проектируемого изделия,

$MLib^{OCC}$ – множество стандартных библиотек Open CASCADE Technology.

Программным образом задавая порядок и условия выполнения процедур из множества $MDes^{OCC}$, формируется упорядоченная последовательность проектных процедур $DesPr^{OCC}$:

$$DesPr^{OCC} = \sum_{i=0}^n \left\{ n_i, dDes_i^{OCC}, dCon_i^{Des.}, dAlt_i^{Des} \right\}, \quad (11)$$

где n_i – порядковый номер выполнения i -й проектной процедуры $dDes_i^{OCC}$ из множества $MDes^{OCC}$, $dCon_i^{Des.}$ – условия выполнения данной процедуры, $dAlt_i^{Des}$ – ветвь альтернативы, к которой принадлежит данная проектная процедура.

$DesPr^{OCC}$ представляет собой древовидную систему проектных маршрутов, обеспечивающую формирование твердотельной 3D-модели проектируемого изделия с учетом исходных и локальных данных, тем самым позволяя получать на выходе класс проектных решений.

Рисунок 9 демонстрирует, что применение технологии Open CASCADE позволяет не только автоматизировать древовидную структуру формирования проектных решений, но и накапливать проектные данные как путем добавления новых ветвей проектных маршрутов, так и модификацией уже имеющихся.

ЗАКЛЮЧЕНИЕ

Представление процесса проектирования системой семантически выраженных проектных стадий ставит сво-

ей целью не просто дублирование стандартных библиотечных алгоритмов проектирования, включая имеющиеся ограничения и условия, а формирование системы проектных процедур $MDes^{OCC}$, основанных на технологии Open CASCADE, адаптированных для проектирования конкретного изделия, определяющих процесс и маршруты проектирования, не допускающих отклонений, не вписывающихся, либо нарушающих формируемое решение. Таким образом, стоит задача выделения множества $MDes^{OCC}$, обеспечивающего получение проектного решения (в том числе класса решений), полностью соответствующего исходным данным.

Рассмотренный способ представления процесса проектирования способен обеспечить обобщение, накопление и модифицирование данных процесса формирования проектных решений, при сохранении структурной целостности и корректности как процесса, так и решения.

Дальнейшее исследование позволит выполнять программную реализацию древовидной структуры проектирования класса объектов с возможностью визуализации влияния каждой проектной стадии на формируемое решение, а также методики анализа готового решения и выбора соответствующего ему проектного маршрута.

СПИСОК ЛИТЕРАТУРЫ

1. Pokhil'ko A.F., Maslyantcin A.A., Udovichenko A.V., Kuprianov A.A. Formal Representation of Processes of Designing Technical Objects: Integrity Approach, Model and Toolkit // Next Generation Concurrent Engineering: Smart

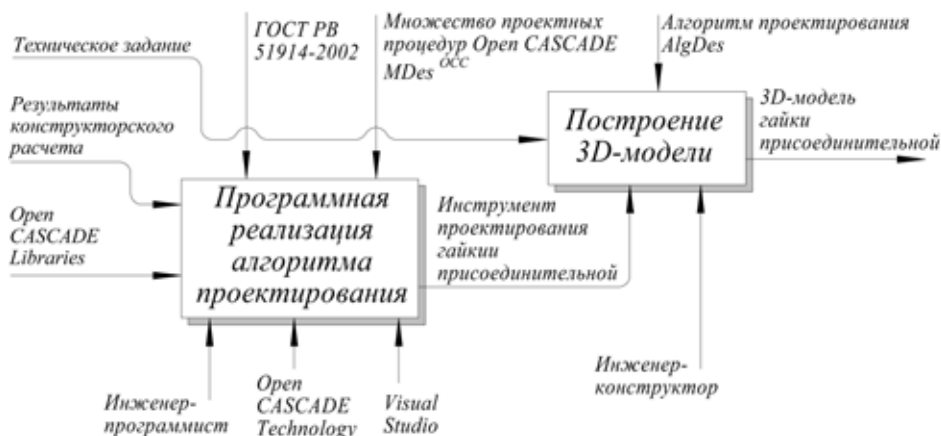


Рис. 9. Автоматизированный процесс построения 3D-модели гайки присоединительной

and Concurrent Integration of Product Data, Services, and Control Strategies, CE 2005, pp. 539–542.

2. Загорулько Ю.А. Технологии разработки интеллектуальных систем, основанные на интегрированной модели представления знаний // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS - 2013): Материалы III Международной научно-технической конференции. – 2013. – С. 31–42.

3. Белецкая С.Ю. Моделирование и поиск оптимальных решений при проектировании сложных систем. – Воронеж : ВГТУ, 2005. – 175 с.

4. Похилько А.Ф., Масляницын А.А., Удовиченко А.В., Куприянов А.А. Формализация и анализ процессов проектирования технических объектов // Автоматизация процессов управления. – 2006. – № 2. – С. 132–140.

5. Kamara J.M., Anumba C.J., and Evbuomwan N.F.O. Process Model for Client requirements processing in construction. *Business Process Management Journal*. 2000. № 6 (3), pp. 251–279.

6. Джурицкий К.Б. Все радиочастотные соединители мира. Ч. 2. Большие, средние и миниатюрные соединители // Компоненты и технологии. – 2012. – № 11. – С. 33–38.

7. ГОСТ 20265-83. Соединители радиочастотные коаксиальные. Присоединительные размеры. – М., 1984. – 14 с.

8. Kim J., Han S. Encapsulation of Geometric Functions for Ship Structural Cad Using a Step Database as Native Storage // *Computer-Aided Design*. 2003. V. 35. no. 13. pp. 1161–1170.

9. O. Kozintsev, A. Pokhilko, L. Kamalov, I. Gorbachev, D. Tsygankov. A Model for Storing and Presenting Design Procedures in a Distributed Service-oriented Environment // *Moving Integrated Product Development to Service Clouds in the Global Economy: Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering*, 2014, pp. 84–91.

10. Васильев К.К., Крашенинников В.Р. Статистический анализ многомерных изображений. – Ульяновск : УлГТУ, 2007. – 170 с.

11. Горбачев И.В., Похилько А.Ф. Технология представления модели в функционально адаптированной САПР // Автоматизация процессов управления. – 2008. – № 3. – С. 39–42.

12. Похилько А.Ф. Формальная система фиксации и обобщения процессов проектной деятельности // Автоматизация процессов управления. – 2010. – № 4. – С. 50–56.

REFERENCES

1. Pokhilko A.F., Maslyantcin A.A., Udovichenko A.V., Kupriyanov A.A. Formal Representation of Processes of Designing Technical Objects: Integrity Approach, Model and Toolkit. *Next Generation Concurrent Engineering: Smart and Concurrent Integration of Product Data, Services, and Control Strategies*, CE 2005, pp. 539–542.

2. Zagorulko Iu.A. Tehnologii razrabotki intellektualnykh sistem, osnovannyye na integrirovannoi modeli predstavleniia

znanii [Technologies of Development of Intelligent Systems Based on the Integrated Knowledge Representation Model]. *Otkrytye semanticheskie tekhnologii proektirovaniia intellektualnykh sistem (OSTIS-2013): Materialy III Mezhdunarodnoi nauchno-tekhnicheskoi konferentsii* [Materials of the Third International Sci. Conf. OSTIS-2013: Open Semantic Technologies for Intelligent Systems], 2013, pp. 31–42.

3. Beletskaiia S.Iu. *Modelirovanie i poisk optimalnykh reshenii pri proektirovanii slozhnykh sistem* [The Modeling and Search of Optimal Solutions in the Complex Systems Developing]. Voronezh, VSTU Publ., 2005. 175 p.

4. Pokhilko A.F., Maslyanitsyn A.A., Udovichenko A.V., Kupriyanov A.A. Formalizatsiia i analiz protsessov proektirovaniia tekhnicheskikh obektov [The Formalization and Analysis of Technical Objects Engineering Processes]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes]. 2006, no.2, pp. 132–140.

5. Kamara J.M., Anumba C.J., and Evbuomwan N.F.O. Process Model for Client Requirements Processing in Construction. *Business Process Management Journal*, 2000, no. 6 (3), pp. 251–279.

6. Dzhurinskii K. Vse radiochastotnye soediniteli mira. Ch. 2. Bolshie, srednie i miniatiurnye soediniteli [All Radio-Frequency Connectors of the World. Part 2. Large, Middle, and Subminiature Size Connectors]. *Komponenty i tekhnologii* [Components and Technologies Magazine], 2012, no. 11, pp. 33–38.

7. GOST 20265-83. *Soediniteli radiochastotnye koaksialnye. Prisoedinitelnye razmery* [Radio-Frequency Coaxial Connectors. Connecting Dimensions]. Moscow, 1984. 14 p.

8. Kim J., Han S. Encapsulation of Geometric Functions for Ship Structural Cad Using a Step Database as Native Storage. *Computer-Aided Design*, 2003, vol. 35, no. 13, pp.1161–1170.

9. Kozintsev O., Pokhilko A., Kamalov L., Gorbachev I., Tsygankov D. A Model for Storing and Presenting Design Procedures in a Distributed Service-Oriented Environment. *Moving Integrated Product Development to Service Clouds in the Global Economy: Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering*, 2014, pp. 84–91.

10. Vasiliev K.K., Krashennnikov V.R. *Statisticheskii analiz mnogomernykh izobrazhenii* [Statistical Analysis of Multidimensional Images]. Ulyanovsk, ULSTU Publ., 2007. 170 p.

11. Gorbachev I.V., Pokhilko A.F. Tehnologiiia predstavleniia modeli v funktsionalno adaptirovannoi SAPR [A Model Representation Technology in the Function Adapted CAD-System]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes]. 2008, no. 3, pp. 39–42.

12. Pokhilko A.F. Formalnaia sistema fiksatsii i obobshcheniia protsessov proektnoi deiatelnosti [Formal System of Fixation and Generalization of Design-Activity Processes]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2010, no. 4, pp. 50–56.