

УДК 004.942

А. П. Соколов, А. Ю. Першин

Московский государственный технический университет им. Н. Э. Баумана

Система автоматизированного проектирования композиционных материалов. Часть 1. Концепции, архитектура и платформа разработки

Настоящая статья – первая в цикле из трех статей о создании системы автоматизированного проектирования композиционных материалов. Первая посвящена базовым концепциям, архитектурным особенностям системы в целом и описанию платформы разработки; во второй будут представлены вычислительная подсистема и принципы организации распределенных вычислений с применением графоориентированного подхода; третья посвящена применению графоориентированной методологии для разработки средств взаимодействия пользователя с системой. В данной статье представлен широкий обзор существующих технологий и программного обеспечения, применяемых для автоматизации решения задач, возникающих в процессе проектирования композиционных материалов. Обоснована актуальность разработки отечественного программного обеспечения в форме интегрированной системы, определены назначение и требования к системе и ее архитектуре. Представлен формализованный подход к разработке системы, базирующийся на применении компонентно-ориентированной многозвенной клиент-серверной архитектуры, применяемой, как правило, для создания промышленных распределенных программных систем. Введены классификации и принципы именования компонентов системы. Введено понятие «функциональный компонент системы». С целью повышения производительности разработки новых компонентов системы создана и задействована подсистема генерации кода на основе шаблонов. Представлены принципы автоматического распределения прав доступа к функциональным компонентам. Результатом проведенной разработки системы стало формирование платформы разработки SA2, фактический состав которой представлен в статье, как и схема разработанной системы, а также перечислены базовые реализованные подсистемы.

Технологии разработки инженерного программного обеспечения, графоориентированная программная инженерия, разработка вычислительных библиотек, автоматизированное проектирование композиционных материалов, система автоматизированного проектирования, система инженерного анализа, разработка модулей расширения

Применение автоматизированных систем различных классов (CAD/CAE/CAM/PLM) при проектировании новых композитных конструкций и материалов в современных научно-исследовательских и проектных организациях является нормой [1]. Ключевую роль в рассматриваемой области играют системы инженерного анализа (CAE), развитие которых осуществляется более 40 лет. Основными известными системами и их производителями являются: ANSYS (ANSYS Inc., США), SIMULIA/ABAQUS («Dassault Systemes», Франция), NASTRAN («MSC Software», США), Comsol, LS-DYNA, Siemens NX CAE («Siemens», Германия), «Altair Engineering» (США), ESI Group (Франция), LMS Int. (Бельгия) и др.; в том числе открытые: Salome, Code_Aster, Calculix, OpenFOAM, Code_Saturne и др. Среди отечественных CAE-систем мож-

но выделить: APM WinMachine (НТЦ «АПМ»), Логос (РФЯЦ ВНИИЭФ), CAE-Fidesys (ООО «Фидесис»), САПР «Сударушка», ИСПА, в том числе отраслевые: Лира, SimInTech [2].

Задача автоматизированного проектирования композиционных материалов (АП КМ) предполагает необходимость решения многочисленных подзадач (рис. 1), относящихся к термомеханике гетерогенных сред; генерации представительных элементов объема (ПЭО) (TexGen, WiseTex [3]); генерации конечно-элементных аппроксимаций ПЭО (известны открытые генераторы: NetGen [4], Gmsh [5], Triangle и др.) и т. п. [6].

Обладая существенным спектром функциональных возможностей, эффективное использование CAE-систем широкого профиля применения требует отдельной подготовки для их про-

фессионального использования и зачастую не позволяет быстро поставить и решить конкретную прикладную задачу, включая АП КМ. Как правило, для решения каждой из указанных подзадач с использованием таких САЕ-систем необходима разработка специализированных сценариев (программ) на предметно-ориентированных языках (например, ANSYS APDL). Многообразие типов КМ, применяемых для их исследования моделей и вычислительных методов приводит к необходимости создания различных сценариев для решения буквально каждой рассматриваемой подзадачи. Такой подход позволяет эффективно решать научно-исследовательские задачи анализа характеристик конкретных КМ, однако *сильно ограничен для общего случая задачи АП КМ с различными схемами армирования, моделями материалов компонентов в условиях многократного решения задач анализа характеристик КМ, решения обратных задач микромеханики КМ и пр.* Для решения задачи АП КМ в столь общей формулировке инженеру требуется целый комплекс программных инструментов, в том числе ориентированных на использование в данной среде многопроцессорной техники.

Одна из важных тенденций в области развития САЕ-систем в мире – это их укрупнение за счет интеграции в них других специализированных систем в форме подсистем или модулей. Стратегически развитие системы может быть обеспечено: а) с помощью реализации открытых интерфейсов, позволяющих создавать модули расширения функциональных возможностей; б) за счет создания собственных базовых, широко используемых и высокоэффективных библиотек, интегрируемых в чужие системы добровольно,

что со временем будет формировать технологические зависимости, ограничивающие развитие чужих систем. Все производители известных САЕ-систем предоставляют возможности как для написания модулей расширения к их системам, так и для интеграции их конечно-элементных решателей во внешние специализированные САПР, в том числе их в САПР КМ.

На сегодняшний день специализированных САПР КМ немного: доступные, часто интегрированные в известные крупные САЕ-системы в форме подсистем, реализуют частные модели гетерогенных материалов (слоистые, N-D-армированные, некоторые дисперсно-армированные) и обычно предполагают дополнительную оплату лицензии. Известными САПР КМ в определенной степени можно считать: Digimat («MSC Software»), FiberSim («Siemens PLM Software»), Moldex3D (отдельные модули ориентированы на моделирование процессов в КМ), SACOM (Japan, «Masagu Zako») [7], SAMCEF («Samtech», отдельные модули), CADEC [8]. Особо следует отметить активно развивающиеся системы, предоставляющие доступ к своим возможностям согласно принципу «System as a Service»: SimScale, cdmHUB.

Некоторые из указанных систем изначально были разработаны как независимые программные комплексы, но со временем в том или ином виде были интегрированы в крупные мультидисциплинарные системы (Ansys Multiphysics, Siemens PLM и пр.), что увеличивает и без того существенную стоимость их использования. Отечественные полноценные САПР КМ фактически отсутствуют.

Несмотря на указанные сложности эксплуатации и существенную стоимость, а также за неимением конкретных альтернатив, в образова-

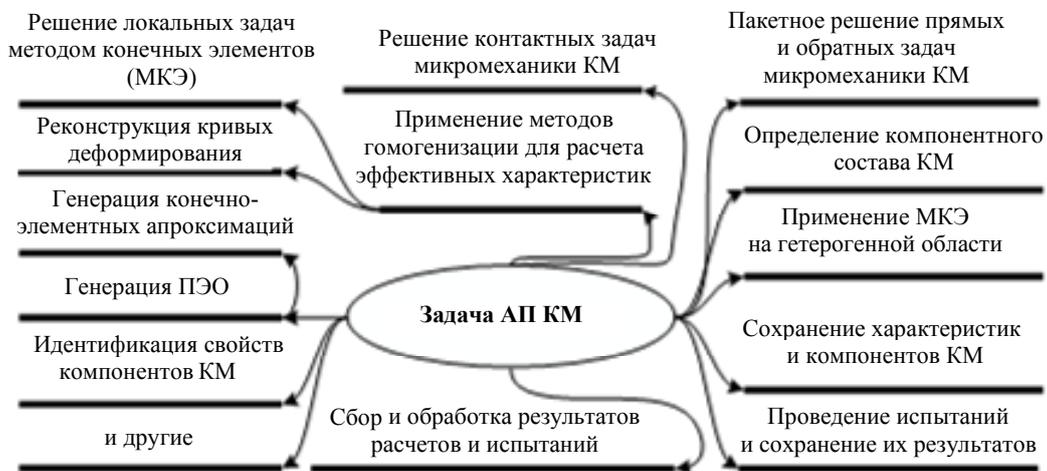


Рис. 1

тельных, научно-исследовательских и проектных организациях в России активно используются в основном именно CAE-системы широкого профиля применения, в том числе при решении задачи АП КМ. Например, в [9], [10] представлен подход к автоматизированному проектированию тканевых КМ с использованием ABAQUS.

Получаемых результатов обычно достаточно для подготовки соответствующих научных публикаций, описывающих опыт использования той или иной модели КМ, что, в свою очередь, позволяет активно развиваться соответствующим CAE-системам и все больше усугубляет положение дел с развитием отечественных САПР.

САПР в общем и САПР КМ в частности относятся к наукоемкому и, как правило, ресурсоемкому программному обеспечению (ПО). Для процесса разработки такого ПО характерны следующие особенности: существенные временные и материальные затраты; серьезные отличия от процесса разработки ПО других классов; обязательное привлечение высококвалифицированных кадров, обладающих междисциплинарными компетенциями как в областях математического моделирования и численных методов, так и в областях механики композиционных материалов и разработки ресурсоемкого ПО специального назначения. Некоторые технические детали разработки наукоемкого и ресурсоемкого ПО освещаются в [11].

Упоминания о работах, проводимых в направлении разработки таких систем, встречаются как в отечественных, так и в зарубежных научных работах (Herworth A. I., Jensen C. G., Roach J. T., 2009 [10], Joseph J. Quigley IV, R. Byron Pipes 1983, Aboudi J. 1989, Baker, P. 1989 [12], Hoа, S. V., Journeaux B. H., Di Lalla, L., 1989 [13], Sun. W., Lau. A., 1997, Kumar. V, Dutta. D., 1998, Sun. W., Lin. F., Hu. X., 2001 [14], Vasey-Glandon V. M., Hale R. D., 2000 [15], Кравченко Ю. А., 2001 [16], Huang Z.-M., Ramakrishna S., 2002, Prodromou A., 2004, Kim .J.-S., 2005 [17], Дмитренко Е. Н. [18], Димитриенко Ю. И., Соколов А. П., 2008 [19], Бобрышев А. Н., Жарин Д. Е., Шафигуллин Л. Н., Гумеров М. И. 2009 [20], Dolšak B., Novak M. [21]).

Среди важных особенностей большинства научных работ и публикаций рассматриваемого направления следует отметить *фактическое отсутствие описания конкретных технических деталей реализации представляемых в них САПР КМ*

[15]–[18], [22]–[25], лишь в некоторых публикациях таковые могут быть найдены [14], [19], [21], [26]–[28]. Достаточно глубоко раскрыта тема особенностей разработки САПР тканевых КМ в [29].

Во многих публикуемых сейчас работах либо описывается опыт использования САПР зарубежного производства, либо представляется лишь поверхностная информация о создаваемой САПР того или иного назначения. Такое положение дел предположительно обусловлено монополизацией рынка CAE-систем, а также глубокой степенью интеграции зарубежных CAD/CAE/CAM/PLM-систем в рамки бизнес-процессов проектных и научно-исследовательских организаций в России.

Несмотря на указанные сложности при создании, использование отечественных САПР при разработке высокотехнологической продукции определяет основу обеспечения национальной безопасности. В перечне поручений по реализации Послания Президента Федеральному Собранию от 24 янв. 2020 г. (Пр-113, п. 1 к) предусмотрено «внести в национальную программу „Цифровая экономика Российской Федерации“ изменения, предусматривающие разработку и внедрение отечественного программного обеспечения и программно-аппаратных средств в целях ускорения цифровой трансформации отраслей реального сектора экономики». Создание САПР в области КМ имеет отношение сразу как минимум к двум критическим технологиям Российской Федерации, утвержденным Указом Президента РФ от 7 июля 2011 г. № 899: *a* – компьютерное моделирование наноматериалов, наноустройств и нанотехнологий; *b* – технологии и программное обеспечение распределенных и высокопроизводительных вычислительных систем.

В результате проведенного анализа может быть сделан вывод об актуальности разработки отечественного наукоемкого ПО в целом и САПР КМ в частности.

Логика изложения материала. Разработка любой системы, в том числе и из рассматриваемого класса, включает несколько этапов: подготовительный (формулировка идеи, определение назначения и требований); проектирование; разработка; тестирование; внедрение и сопровождение.

Заявленная масштабная тематика статьи формально объединяет в своем составе разноплановые аспекты разработки системы в целом, среди которых: описание процессов организации разработки, архитектурные особенности системы и ее

ключевых подсистем, описание применяемых математических моделей, вычислительных методов и особенностей архитектуры программных модулей решения задач в области механики КМ. Очевидно, что объема одной статьи недостаточно для содержательного изложения такого материала, поэтому было принято решение разбить его на отдельные части: концепции, архитектурные особенности и описание платформы разработки (часть 1); вычислительная подсистема, принципы организации распределенных вычислений с применением графоориентированного подхода (часть 2); применение графоориентированной методологии для разработки средств взаимодействия пользователя в системах инженерного анализа (часть 3). С точки зрения авторов, этот материал является ключевым для формирования достаточно полного представления об основах разработанной системы, принципах ее функционирования и некоторых возможностях.

Настоящая статья – первая в указанном цикле, и ее цель – кратко представить созданную за прошедшие годы систему, базовые концепции, платформу разработки, осветить ее ключевое

назначение, реализуемые требования, а также ввести ряд важных понятий. В статье представлены обобщающие результаты работ, выполненных авторами в период с 2009 г. по настоящее время.

Наименование, назначение и требования к системе. Представляемая система автоматизированного проектирования КМ (САПР КМ) является четвертой версией системы GCAD (табл. 1), современное название которой – «Распределенная вычислительная система GCD» (PBC GCD).

Представляемой САПР КМ развивалась постепенно. С включением в состав новых функциональных возможностей исходное назначение системы также несколько модифицировалось, что стало основанием для изменения исходного наименования GCAD на PBC GCD в 2010 г. Далее по тексту будет использоваться обозначение САПР КМ, указывающее на предметное назначение системы.

Назначением текущей версии САПР КМ является решение следующих комплексных задач (далее – классов задач), возникающих в процессе проектирования КМ: *a* – анализ эффективных физико-механических характеристик (ФМХ) КМ

Таблица 1

Наименование САПР КМ, версия	Описание, некоторые функциональные возможности и используемые технологии	Годы разработки
WayList (прототип GCAD v.1)	Автоматизация хозяйственной деятельности транспортной организации. Архитектура: двухзвенная клиент-серверная. Стек технологий: СУБД Interbase, Borland C++, VCL, IDE: C++ Builder	2003–2004
GeometryCAD (GCAD v.1)	Построение ПЭО тканевых КМ с полотняным типом переплетения и расчет эффективных упругих характеристик (ЭУХ) методом асимптотического осреднения. Решение задач напряженно-деформированного состояния (НДС). Архитектура: двухзвенная клиент-серверная. Стек технологий: СУБД Interbase, Borland C++, VCL, IDE: C++ Builder	2004–2005
GeometryCAD v.2 (GCAD v.2)	Решение задач о поиске ЭУХ многоуровневых КМ, построение геометрических моделей ячеек периодичности КМ различных типов (3D, 4D – армированных, тканевых) [19]. Архитектура: двухзвенная клиент-серверная. Стек технологий: СУБД PostgreSQL, MSVC++, QT, aINI [30], стандарт разработки [31], CodeGenerator [32], IDE: Visual Studio 2003 г.	2005–2008
Распределенная вычислительная система GCD v.1, (PBC GCD (ex. GCAD v.3))	Анализ и хранение упруго-прочностных свойств КМ, генерация ПЭО, ведение НИР, построение решателей задач механики КМ. Доступ к функциональным возможностям обеспечен посредством desktop-клиента. Архитектура: трехзвенная клиент-серверная. Стек технологий: СУБД PostgreSQL, gcc, MSVC++, nsis, TCP/IP, QT, XML, XSLT, GUIINIBuilder [30], CodeGenerator [32], aINI [30], стандарт разработки [31], IDE: Visual Studio 2003 г.	2009–2012
Распределенная вычислительная система GCD v.2, PBC GCD v.2 (ex. GCAD v.4)	Реализованы ядро системы и подсистемы: управление проектами, подсистемы: разработчика, инженера-исследователя, инженера-технолога, прикладные подсистемы: вычислительных экспериментов, микромеханики композитов, химико-технологических процессов, оптимизации и др. Доступ к функциональным возможностям обеспечен посредством web- и desktop-клиентов. Базовые библиотеки: парсеры aINI, aDOT, графоориентированный каркас. Функции: более 460. Архитектура: трехзвенная клиент-серверная. Стек технологий: СУБД PostgreSQL, Python, gcc, TCP/IP, Django, XML, XSLT, платформа SA2 (табл. 2), IDE: любая	С 2013 по наст. вр.

различных типов; \bar{b} – уточнение ФМХ компонентов проектируемых КМ; \bar{v} – определение компонентного состава проектируемых КМ; \bar{z} – реконструкция эффективных кривых деформирования и поверхностей прочности проектируемых КМ; \bar{d} – сбор и обработка результатов расчетов и испытаний.

Каждая из указанных задач в общем случае включает многие подзадачи. Например, среди анализируемых ФМХ проектируемых КМ актуален анализ: упругих, тепловых, термоупругих, прочностных, термопрочностных, вязкоупругих, электромагнитных и прочих характеристик. Задача уточнения ФМХ компонентов проектируемого КМ предполагает постановку и решение обратных задач микромеханики КМ [33], которые предполагают многократное решение прямых задач, каждая из которых обычно подразумевает применение методов гомогенизации [34], [35] и метода конечных элементов.

Разнообразие подзадач, которые возникают и могут возникнуть в процессе решения комплексной задачи АП КМ в общей постановке, привело к необходимости создания систематизированного подхода к их решению. Большинство среди перечисленных задач предполагают применение различных, в том числе ресурсоемких, вычислительных методов, поэтому должен был быть создан единый подход к разработке высокоэффективных программных реализаций последних [36].

После обобщения особенностей необходимых для решения задач были сформулированы основные требования к свойствам будущей системы в целом: расширяемость; масштабируемость; кроссплатформенность; унификация компонентов системы; возможность коллективной разработки и гибкого сопровождения системы в целом, а также ее подсистем и отдельных модулей; поддержка единых форматов входных данных (в том числе известных); обеспечение единого алгоритма согласования взаимозависимых вычислительных методов, реализуемых в системе.

Базовые принципы разработки. Важный принцип, применяемый при создании крупных систем, – стандартизация и регламентация процессов разработки (именование компонентов, оформление кода, структуры каталогов, принципы маркировки при редактировании технических документов и пр.). Собственный стандарт был разработан еще в 2006 г. и с некоторыми модификациями используется до сих пор [31], в том числе при создании представляемой САПР КМ. Помимо указанного к базовым принципам создания САПР КМ следует отнести:

- использование системной базы данных (регистрация подсистем, функций, решателей задач, пользователей, прав доступа и т. п.);
- введение классификаций компонентов системы (по назначению, по методам реализации и т. п.);
- использование единых принципов именования компонентов системы;
- применение средств автоматизации процесса разработки (инструменты генерации кода, системы контроля версий, облачные сервисы).

Далее по тексту используются специальные обозначения (табл. 2).

Компоненты системы и их классификации. Любая система состоит из подсистем, а они, в свою очередь, – из программных модулей. Каждый программный модуль может быть реализован по-разному: в форме одного или нескольких файлов в заранее определенных форматах.

Компонентами программной системы в общем случае служат:

- программные компоненты: подсистемы, исполняемые программные модули, модули расширения, библиотеки, файлы настроек, шаблоны файлов исходных данных, файлы ресурсов, форматы данных, структуры данных, классы C++, программные документы и т. д.;
- компоненты баз данных: таблицы, домены, схемы данных, представления данных, хранимые процедуры, триггеры, и т. д.

Таблица 2

Обозначение	Описание
@P@	«Скобки» такого вида используются для обозначения параметра P , правила определения которого задаются отдельно (при подстановке значения их следует удалять вместе с именем параметра)
char(N)	Обозначение строкового типа, объекты которого должны состоять ровно из N символов
INd(T)	Множество всевозможных объектов типа T
а) lc(N), б) uc(N), в) UCC($< N$), г) ICC($< N$)	Обозначения форматов, которые определяют требования при подготовке строковых идентификаторов длиной N или не более N ($< N$): а) использование нижнего (lower case) регистра; б) использование верхнего (UPPER CASE) регистра; в) в одно слово без пробелов, каждое слово с большой буквы (UpperCamelCase); г) в одно слово без пробелов, каждое слово, кроме первого, с большой буквы (lowerCamelCase)

В связи с отсутствием априорной информации о количестве и назначении компонентов, которые потребуются для создания САПР КМ, возникла необходимость разработки такой архитектуры системы, которая бы обеспечила гибкое расширение ее функциональных возможностей в перспективе. Очевидным решением стало введение и использование специальных систем классификации всех составных частей (компонентов) будущей системы, а также создание системной базы данных (БД) для регистрации: компонентов, их различных типов, классов, форматов данных и прочих системных атрибутов. Принципиальная UML-схема классов компонентов системы и их классификаций представлена на рис. 2. Штриховой линией обозначены необязательные (слабые) связи. В скобках указаны наименования таблиц системной базы данных, в которых регистрируются соответствующие объекты.

Системная база данных была разработана с учетом неопределенности исходных требований к составу компонентов будущей системы, при ее создании использовалась специальная нотация именования ее компонентов [31]: таблиц, доненов, хранимых процедур и пр. Особенности ее разработки выходят за рамки настоящей статьи.

Одной из основных в системе является *трехуровневая иерархическая классификация «комплекс-решение-проект»*, которая обеспечивает возможность связать конкретный компонент с той или иной частью системы. Классификация определяется тремя множествами: «комплексами» C , «решениями» $S(c)$ и «проектами» $P(c, s)$. «Комплексы» состоят из «решений», а «решения», в свою очередь, состоят из «проектов» (рис. 2):

$$C = \{c : c \in \text{Ind}(\text{char}(3))\},$$

$$S(c) = \left\{ \begin{array}{l} \text{pr}_1(s_j) \#\#\text{pr}_2(s_j) : s_j \in C \times \text{Ind}(\text{char}(3)), \\ \text{pr}_1(s_j) = c \in C, \\ \forall s_1, s_2 : s_1 \neq s_2 \Rightarrow \text{pr}_2(s_1) \neq \text{pr}_2(s_2), \end{array} \right\}$$

$$P(c, s) = \left\{ \begin{array}{l} \text{pr}_1(p_k) \#\#\text{pr}_2(p_k) : \\ p_k \in S(c) \times \text{Ind}(\text{char}(3)), \\ \text{pr}_1(p_k) = s \in S(c), c \in C, \\ \forall p_1, p_2 : p_1 \neq p_2 \Rightarrow \text{pr}_2(p_1) \neq \text{pr}_2(p_2), \end{array} \right\}$$

где $\text{pr}_i(y)$ – проекция i от y ; $A\#\#B$ – конкатенация строк A и B .

Примеры конкретных множеств: $C = \{\text{'gcd'}, \text{'chm'}, \dots\}$ – «комплексы»; $S(c) = \{\text{'gcdfes'}, \text{'gcdgpr'}, \dots\}$ – «решения» комплекса 'gcd'; $P(c, s) = \{\text{'gcdfesmd1'}, \text{'gcdfesmd2'}, \dots\}$ – «проекты».

При разработке программного обеспечения под «комплексами» будем понимать системы, под «решениями» – подсистемы, а под «проектами» – программные модули и/или проекты по их разработке.

При создании компоненты обязательно связываются с конкретными «комплексом» и «решением», и дополнительно может быть определена необязательная связь с «проектом» (рис. 2). Все предметно-ориентированные компоненты представляемого САПР КМ принадлежат «комплексу» 'gcd'; ядро системы определяется компонентами, входящими в состав ряда системных «комплексов» $c \in \{\text{'sys'}, \text{'com'}, \dots\}$.

Вторая важная классификация компонентов в САПР КМ – классификация по предметному

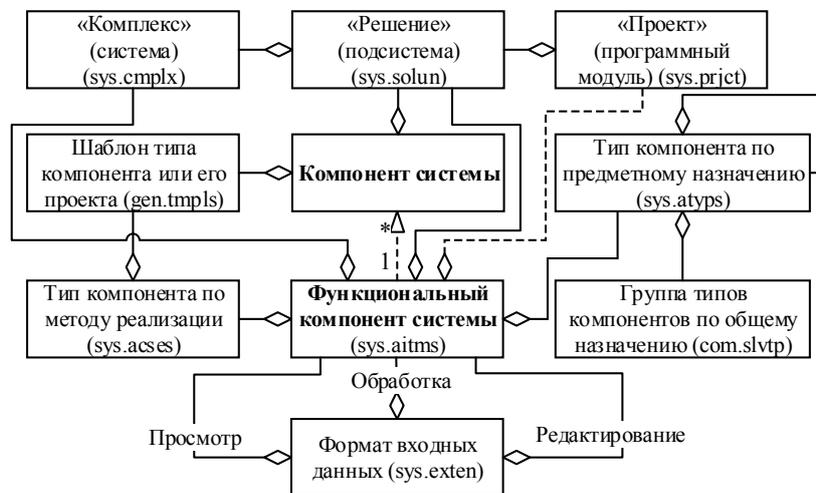


Рис. 2

назначению (рис. 2), определяемая следующим множеством:

$$Y = \left\{ y : y \in \text{Ind}(\text{char}(15)) \times \text{Ind}(\text{char}(15)) : \right. \\ \left. \forall y \Rightarrow !\exists p \in Y : \text{pr}_2(y) = \text{pr}_1(p) \right\}$$

Каждый компонент системы разрабатывается для решения определенной задачи, а каждую из решаемых задач следует относить к заранее заданному классу решаемых задач, которому в рамках разрабатываемой САПР КМ соответствует тип компонентов по предметному назначению. В свою очередь, каждый тип компонентов по предметному назначению может иметь родительский более общий тип. Например, тип «функции генерации тканевых ячеек периодичности» дочерний по отношению к родительскому: «функции генерации представительных элементов объема».

Большинство задач, решаемых в программных системах, требуют разработки похожих «внешне» и по назначению элементов. Например: формы ввода-вывода, функции представления данных в табличном и/или графическом виде. Третья важная введенная и используемая в системе классификация компонентов – это классификация по методу реализации, определяемая множеством методов $M = \{m : m \in \text{Ind}(\text{char}(15))\}$. Данная классификация необходима для того, чтобы определить методы расширения функциональных возможностей системы. За счет ограниченного набора методов процесс разработки крупной системы может быть существенно упрощен.

В [30] представлен программный подход по генерации графических пользовательских интерфейсов ввода на основе интерпретации файлов входных данных, определяемых с использованием языка aINI. Рассматриваемый подход служит программной реализацией одного из таких стандартных методов реализации компонентов:

Важной абстракцией в системе является понятие «функциональный компонент».

Определение 1. Функциональным компонентом (англ. *Action Item (AI)*) будем называть обоб-

щенную именованную запускаемую функциональную возможность, заранее зарегистрированную в системе либо обладающую свойством динамической регистрации среди доступных функциональных возможностей во время выполнения, ассоциированную с конкретными: «комплексом», «решением», «проектом», типами по предметному назначению и методу реализации.

Замечания. 1. Реализация функционального компонента также определяется ассоциированным с ней методом реализации.

2. Функциональные компоненты могут быть зарегистрированы в системной базе данных и быть общедоступными, а могут в ней отсутствовать и быть скрытыми, динамически регистрируемыми в системе при ее запуске.

Единый формат именования компонентов. С увеличением количества компонентов системы, очевидно, возникает задача их именования. С одной стороны, имя должно быть уникальным, а с другой – ассоциироваться с конкретной системой, подсистемой, явно или неявно указывать на метод, с помощью которого этот компонент был разработан, а также, возможно, на тип компонента. Указанное требование может быть реализовано только в случае использования строковых идентификаторов, имеющих явную связь с соответствующими типами компонентов. Введен единый формат именования компонента: @cpxsln@_@tml@_@Name@, где @cpxsln@, @tml@ – строковые идентификаторы «решения» и шаблона в форматах lc(6) и lc(3) соответственно; @Name@ – имя компонента в формате UCC(<25). На основе указываемого шаблона может быть создана заготовка компонента определенного типа [32]. Примеры имен представлены в табл. 3.

Замечание. Отметим, что не все компоненты следует именовать представленным стандартным образом. Если компонент используется исключительно в объектах «проекта», в рамках которого он разрабатывается, то ему может быть дано имя в произвольном формате. В указанном случае разработчик должен осознавать, что использование этого компонента в других «решениях» не предполагается.

Таблица 3

Компонент	Примеры имен	Компонент	Примеры имен
Файл исходного кода	gcdfes_unt_MySourceCode.cpp	Файл исходных данных	gcdgui_plg_LogHandler.aini
Программный документ	educon_evn_ICMMT.pdf	Библиотека	gcdlog_dll_LogEventCore.dll
Файл графовой модели	gcdrmh_dot_ElstFemSolver.adot	Презентация	edumms_lec_Introduction.pdf
Имя «комплекса» (системы, схемы данных БД)	rnd, sys, gcd, edu	Имя «решения» (репозитория или подсистемы)	gcdrmh_rndhpc, comsdk, edurk6

Форматы имен «комплексов», «решений», git-репозиторийев, схем данных, таблиц БД и ряда прочих объектов, имена которых следует определять иначе, представлены в [31].

Применение средств автоматизации процесса разработки. К числу важных программных технологий, используемых при создании САПР КМ, следует отнести вспомогательное программное обеспечение генерации кода на основе шаблонов. Такие системы принято относить к средствам поддержки процесса разработки (CASE-инструментариям). В процессе разработки САПР КМ была создана специальная подсистема генерации кода на основе шаблонов [32], которая вошла в состав программной платформы разработки SA2 (табл. 4). Для регистрации шаблонов различных стандартных компонентов системы и для хранения параметров генерации используются таблицы в системной базе данных. Непосредственное хранение шаблонов осуществляется в репозиториях системы контроля версий Git.

Замечание. Использование генератора кода обеспечивает автоматизацию построения проектов компонентов, их размещение в репозиториях систем контроля версий, интеграцию с системами непрерывной интеграции, а также генерацию имен компонентов согласно заранее задаваемому формату.

Автоматическое распределение прав доступа. Объектом в системе, в отношении которого должны быть определены права доступа, является функциональный компонент (AI). Явное определение прав доступа к конкретному AI конкретному пользователю крайне затруднительно, так как заранее неизвестны ни списки пользователей, ни перечни компонентов.

Распределение прав доступа к AI, которые еще не созданы, обеспечивается посредством определения прав доступа к типу компонентов по *предметному назначению*. В результате ко всем компонентам данного типа будут определены соответствующие права доступа.

Принцип расширения функциональных возможностей. Использование представленных ранее технологий позволяет различным специалистам независимо друг от друга вести разработку функциональных компонентов и автоматически интегрировать их в единую систему.

Расширение функциональных возможностей системы было реализовано созданием менеджера плагинов различных типов, принцип работы которого представлен далее.

Клиентское приложение посредством сервера приложений обращается к системной БД и, в зависимости от прав подключающегося пользователя, определяемых его ролью (ролями), загружает информацию о доступных этому пользователю типах компонентов по предметному назначению. На основе полученного списка предметных типов формируется «пустое» дерево функциональных возможностей без «листьев». Каждый реализуемый AI связан с конкретным типом AI по предметному назначению, что обеспечивает возможность их автоматического включения в качестве «листа» в соответствующую ветку построенного дерева. Включение конкретного AI в состав дерева функциональных возможностей зависит: а) от прав доступа пользователя к узлу дерева; б) типа метода реализации данного AI (включаться могут только такие AI, которые могут быть запущены); в) значения атрибута «скрытый AI» и ряда прочих.

Таблица 4

Наименование элемента	Описание
Стандарт разработки	Документ, регламентирующий общие принципы ведения разработки [31] (применяемые нотации, именованя, методы разработки, общие подходы)
GCDDDB	Системная база данных
Язык aINI	Язык определения входных данных [37]
Язык aDOT	Язык определения графовых моделей сложных вычислительных методов (СВМ) [38]
Формат XDBT	XML-ориентированный формат определения структуры и данных таблицы базы данных
CodeGenerator (gcdcse)	Генератор кода на основе шаблонов [32]
GUIBuilder	Генератор графических пользовательских интерфейсов [30]
comaps, comfrm	Сервер приложений и его базовая библиотека
comwpc	Web-клиент к системе
comsdk (C++) русcomsdk (Python)	Системные библиотеки. Обеспечивают использование единых форматов представления входных данных (aINI) и графовых моделей вычислительных методов (aDOT) в рамках графоориентированного подхода [36]
gcdtml	Библиотека шаблонов компонентов системы
gcddot	Библиотека графовых моделей решателей задач
gcdini	Библиотека входных данных (постановок) задач

Платформа разработки. Свойство расширения функциональных возможностей было заложено в систему в момент ее создания, что позволило обеспечить поэтапный процесс ее развития: каждая новая версия системы наследовала функциональные возможности предыдущей и включала в свой состав все новые программные подсистемы и программные модули.

При реализации каждой новой версии системы постепенно оформлялись ключевые базовые компоненты и библиотеки, которые в результате были выделены в платформу разработки, получившую название SA2. Основные элементы платформы представлены в табл. 4.

Архитектура системы. В основе системы лежит трехзвенная клиент-серверная архитектура, реализующая логические уровни САПР КМ (рис. 3).

Замечание. Отделение уровня данных от уровня пользователя позволяет при необходимости заменить выбранную СУБД, тогда как реализация в том числе прямого доступа между уровнями пользователя и данных позволяет снижать нагрузку на сервер приложений. Отделение уровня пользователя позволяет создавать средства ввода-вывода данных независимо от вычислительных библиотек, разрабатываемых в рамках логического уровня.

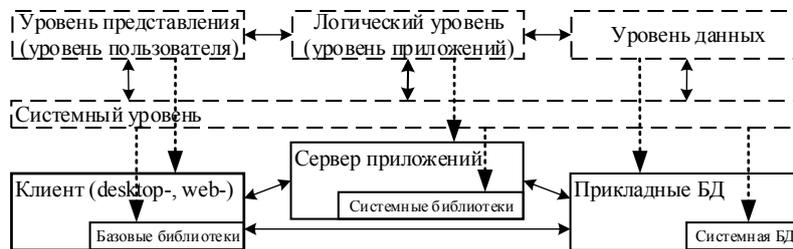


Рис. 3

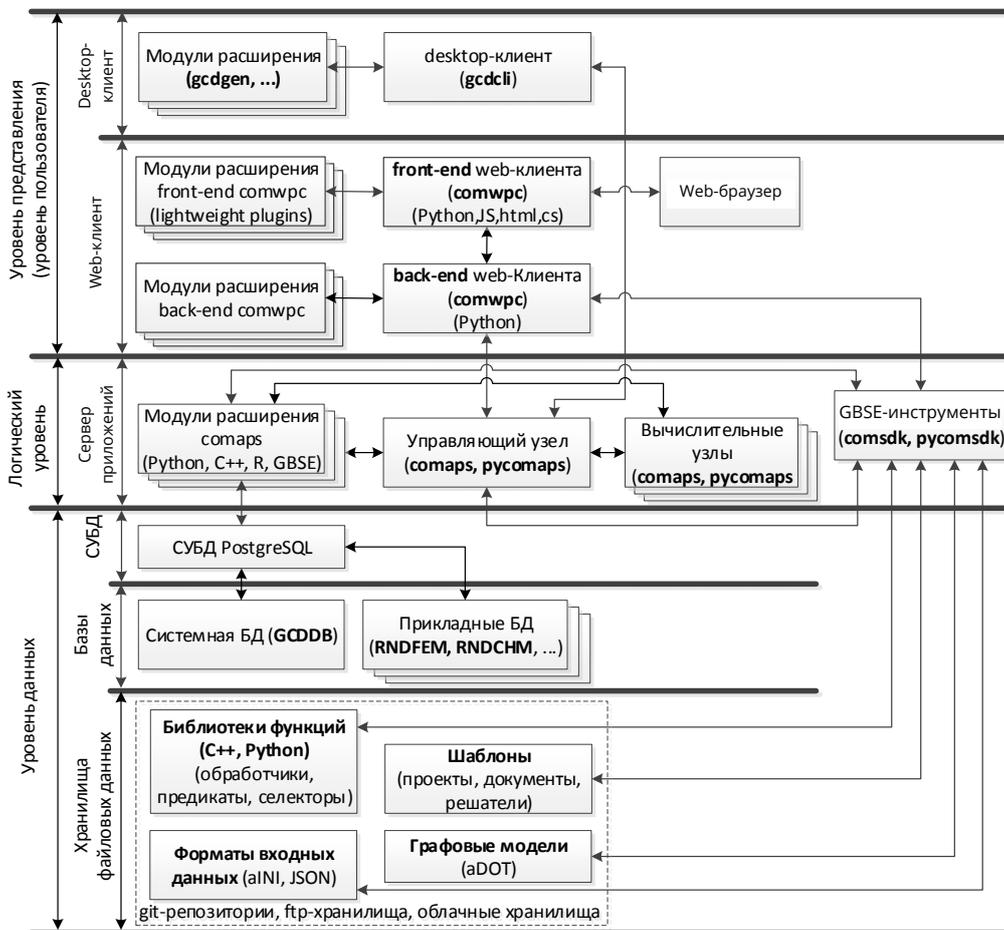


Рис. 4

Детализированная архитектура разработанной системы представлена на схеме (рис. 4).

Desktop-клиент и сервер приложений разрабатывались на языке программирования C++, а web-клиент – с использованием программного каркаса Django с применением языка программирования Python. Системная и прикладные базы данных были разработаны с использованием СУБД PostgreSQL. Для ведения разработки используются: а) система контроля версий Git; б) система управления проектами Redmine. Информационная инфраструктура, применяемая при разработке, соответствует описанной в [39].

К основным прикладным подсистемам разработанного прототипа САПР КМ следует отнести: а) вычислительную подсистему, построенную на основе графоориентированного подхода [36], в рамках которой помимо прочих реализован метод реверсивной многомасштабной гомогенизации [40]; б) подсистему построения представительных элементов объема; в) подсистему сбора характеристик материалов, полученных численно и экспериментально; г) подсистему генерации объектов на основе шаблонов [32]; и ряд других.

Деятельность в направлении развития представленной САПР КМ продолжается в настоящий момент. Работоспособность системы подтверждена полученными результатами при реализации серии научных исследований в области микромеханики композиционных материалов [41]–

[45]. Применение системы обеспечило автоматизацию процессов решения различных задач «пре-процессинга», обработки и «постпроцессинга» данных в области микромеханики КМ, в том числе за счет предоставления удаленного доступа к функциональным компонентам различного назначения: а) построение представительных элементов объема КМ различных типов; б) проведение анализа упругих, тепловых, упруго-прочностных и прочих характеристик КМ; в) визуализация алгоритмов решения расчетных задач; г) разработка научно-технической документации по решателям и пр.

В следующих статьях будут представлены: вычислительная подсистема, результаты ее использования на конкретных примерах решенных прикладных задач, а также методология автоматизированного проектирования КМ, основанная на применении созданного авторами графоориентированного подхода.

Разработка архитектуры САПР КМ, прототипов системы и программной платформы SA2, осуществлена авторами в МГТУ им. Н. Э. Баумана на инициативной основе.

Отдельные подсистемы САПР КМ создавались в разные годы при следующей финансовой поддержке: госзадание 1.6260.2011, гранты Президента РФ МК-6421.2012.9, МК-765.2012.8, МК-6573.2013.3, ФЦП «Кадры» 14.В37.21.1869, ФЦП «Исследования и разработки» 14.577.21.0135, 14.574.21.0158.

СПИСОК ЛИТЕРАТУРЫ

1. Sapuan S. M., Mansor M. R. Concurrent engineering approach in the development of composite products: A review // *Materials and Design*. 2014. Vol. 58. P. 161–167.
2. Компьютерный инжиниринг CAE // Портал УрФУ. URL: <https://cae.urfu.ru/ru/caerus> (дата обращения 22.02.2020).
3. Verpoest I., Lomov S. V. Virtual textile composites software Wisetex: integration with micro-mechanical, permeability and structural analysis // *Compos. Sci. Technol.* 2005. Vol. 65, № 15–16. P. 2563–2574.
4. Schöberl J. NETGEN – An advancing front 2D/3D-mesh generator based on abstract rules. // *Computing and Visualization in Science*. 1997. Vol. 1, № 1. P. 41–52.
5. Geuzaine C., Remacle J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities // *Intern. J. for Numerical Methods in Engineering*. 2009. Vol. 79, № 11. P. 1309–1331.
6. Яблочников Е. И., Васильков С. Д., Фомина Ю. Н. Интегрированные технологии проектирования изделий из полимерных композиционных материалов //

Науч.-техн. вестн. информационных технологий, механики и оптики. 2012. Т. 78, № 2. 109–113.

7. Recent developments in «COMPOSITES DREAM» for meso-FE modelling of advanced materials / T. Sakakibara, T. Yamamoto, G. Yamada, A. Imaoku, R. Ohtagaki, M. Zako // 13th Intern. Conf. on Textile Composites (TEXCOMP-13). IOP Conf. Series: Materials Science and Engineering. 2018. Vol. 406, № 012027. URL: <https://iopscience.iop.org/article/10.1088/1757-899X/406/1/012027> (дата обращения: 03.11.2020).

8. Cosso F. A., Barbero E. J. Computer aided design environment for composites // *Proc. 2012 SAMPE Conf. and Exhibition, Baltimore, USA, May 21–24, 2012*. URL: <http://barbero.cadec-online.com/papers/2012/conference/1987.pdf> (дата обращения 03.11.2020).

9. Kim B. C., Weaver P. M., Potter K. Computer aided modelling of variable angle tow composites manufactured by continuous tow shearing // *Composite Structures*. 2015. Vol. 129. P. 256–267.

10. Hepworth A. I., Jensen C. G., Roach J. T. A CAD independent approach to automate laminate composite design and analysis // *Computer-Aided Design and Applications*. 2009. Vol. 6, № 2. P. 147–156.
11. Особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем / С. В. Ковальчук, С. В. Иванов, И. И. Колыхматов, А. В. Бухановский // *Информационно-управляющие системы*. 2008. № 3. С. 10–18.
12. Baker P. Integrated approach to finite element analysis of advanced composite structures // *Computer-Aided Design*. 1989. Vol. 21, № 7. P. 441–446.
13. Hoa S. V., Journeaux B. H., Di Lalla L. Computer aided design for composite structures // *Composite Structures*. 1989. Vol. 13. P. 67–79.
14. Sun W., Lin F., Hu X. Computer-aided design and modeling of composite unit cells // *Composites Science and Technology*. 2001. Vol. 61, № 2. P. 289–299.
15. Hale R. D., Vasey-Glandon V. M. PACKS: An affordable knowledge-driven composite design for manufacturing process // *Proc. of 46th Intern. SAMPE Symp. and Exhibition*. Long Beach, CA, USA, 2001. Vol. 46 II. P. 1885–1898.
16. Кравченко Ю. А. Разработка и исследование алгоритмов функционирования подсистем САПР оптимизации выбора конструкционных композиционных материалов по критериям прочности: дис. ... канд. техн. наук. Таганрог, 2001.
17. Kim J.-S. Development of a user-friendly expert system for composite laminate design // *Composite Structures*. 2005. Vol. 79. № 1. P. 76–83.
18. Дмитренко Е. Н. Автоматизация проектирования оптимальных составов композиционных материалов промышленного назначения: дис. ... канд. техн. наук. Омск, 2006.
19. Димитриенко Ю. И., Соколов А. П. Система автоматизированного прогнозирования свойств композиционных материалов // *Информационные технологии*. 2008. № 8. С. 31–38.
20. Система автоматизированного проектирования полимерных наполненных композиционных материалов специального назначения. Кузнечно-штамповочное производство / А. Н. Бобрышев, Д. Е. Жарин, Л. Н. Шафигуллин, М. И. Гумеров // *Обработка материалов давлением*. 2009. № 8. С. 9–16.
21. Dolšak B., Novak M. Intelligent decision support for structural design analysis // *Advanced Engineering Informatics*. 2011. Vol. 25, № 2. P. 330–340.
22. Концепция создания системы автоматизированного проектирования процессов производства композиционных материалов (САПР ПКМ) из отходов металлообработки / С. И. Пестрецов, М. В. Соколов, А. Н. Колодин, В. Г. Однолько // *Вопр. совр. науки и практики*. 2011. Т. 32, № 1. С. 386–390.
23. Абдуллин И. А., Гумеров А. Ф., Шафигуллин Л. Н. Автоматизированная информационная система прогнозирования свойств полимерных композиционных материалов на основе регрессионного анализа // *Вестн. Казанского технол. ун-та*. 2012. Т. 15, № 16. С. 240–243.
24. Разработка автоматизированной системы проектирования технологических процессов изделий из композиционных материалов / С. А. Богуславский, А. А. Ларин, В. И. Резниченко, Д. В. Сизов // *Прогрессивные технологии и процессы: сб. науч. ст. 2-й Междунар. молодежной науч.-практ. конф. (24–25 сент. 2015 г.)*. Т. 1. Юго-Зап. гос. ун-т. Курск: ЗАО «Университетская книга», 2015. С. 149–151.
25. Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А. Архитектура распределенных сервис-ориентированных систем автоматизированного проектирования // *Изв. ЮФУ. Технические науки*. 2014. № 7 (156). С. 51–58.
26. Киселев А. М. Прогнозирование механических характеристик 3D-тканей с применением системы проектирования «ПРЕФОРМА» // *Технологии и качество*. 2019. № 2 (44). С. 20–25.
27. Ахмедзянов Д. А., Кишалов А. Е., Маркина К. В. Автоматизированное проектирование авиационных газотурбинных двигателей и выбор материалов их основных деталей // *Вестн. Самарского гос. аэрокосмического ун-та им. академика С. П. Королева (Нац. исслед. ун-та)*. 2015. Т. 14, № 1. С. 101–111.
28. Яблочников Е. И., Восоркин А. С., Цупиков А. В. Интегрированная система для разработки изделий из полимерных композиционных материалов на основе методологии PLM // *Программные продукты и системы*. 2014. № 2. С. 100–104.
29. Meso-FE modelling of textile composites: Road map, data flow and algorithms / S. V. Lomov, D. S. Ivanov, I. Verpoest, M. Zako, T. Kurashiki, H. Nakai, S. Hirose // *Composites Science and Technology*. 2007. Vol. 67. P. 1870–1891.
30. Соколов А. П., Першин А. Ю. Программный инструментарий для создания подсистем ввода данных при разработке систем инженерного анализа // *Программная инженерия*. 2017. Т. 8, № 12. С. 543–555.
31. Соколов Ал. П., Соколов Ан. П. Стандарт разработки программной платформы SA2 // *Облачный сервис SA2 Systems*. URL: <https://sa2systems.ru/nextcloud/index.php/f/403367> (дата обращения 05.03.2020).
32. Разработка программного обеспечения генерации кода на основе шаблонов при создании систем инженерного анализа / А. П. Соколов, В. М. Макаренко, А. Ю. Першин, И. С. Лаишевский // *Программная инженерия*. 2019. Т. 10, № 9–10. С. 400–416.
33. Ватульян А. О. Обратные задачи в механике деформируемого твердого тела. М.: Физматлит, 2007.
34. Бахвалов Н. С., Панасенко Г. П. Осреднение процессов в периодических средах. М.: Наука, 1984.
35. Buryachenko V. A. *Micromechanics of Heterogeneous Materials*. Springer, 2007.
36. Соколов А. П., Першин А. Ю. Графоориентированный программный каркас для реализации сложных вычислительных методов // *Программирование*. 2019. Т. 47, № 5. С. 43–55.

37. Соколов А. П. Описание формата данных aINI (advanced INI) // Облачный сервис SA2 Systems. URL: <https://sa2systems.ru/nextcloud/index.php/f/403527> (дата обращения 05.03.2020).

38. Соколов А. П., Першин А. Ю. Описание формата данных aDOT (advanced DOT) // Облачный сервис SA2 Systems. URL: <https://sa2systems.ru/nextcloud/index.php/f/403526> (дата обращения 05.03.2020).

39. Sokolov A. P., Pershin A. Yu., Vocharov V. A. Integration of open-source software for automation of electronic document flow in the structural unit of an educational organization // IV Intern. Conf. on Information Technologies in Engineering Education (Inforino). Russia, Moscow, 2018. P. 1–4.

40. Sokolov A. P., Pershin A. Yu. Computer-aided design of composite materials using reversible multiscale homogenization and graph-based software engineering // Key Engineering Materials. 2018. Vol. 779. P. 11–18.

41. Численное определение эффективных упруго-прочностных характеристик композитных заделок газоразделительных мембранных модулей / А. П. Соколов, К. В. Михайловский, В. Н. Щетинин, А. С. Сапелкин, В. В. Пресняков // Материалы IX Междунар. конф. по неравновесным процессам в соплах и струях (NPNJ'2016). Россия, Алушта: МАИ, 2016. С. 387–389.

42. Ресурсоемкое численное моделирование упруго-прочностных свойств модифицированных неразъемных клеевых соединений / А. П. Соколов, А. В. Игнатов, В. Н. Щетинин, А. С. Сапелкин // Материалы IX Междунар. конф. по неравновесным процессам в соплах и струях (NPNJ'2016, Россия, Алушта: МАИ, 2016. С. 464–466.

43. Соколов А. П., Щетинин В. Н., Сапелкин А. С. Применение технологии GBSE для разработки программного обеспечения автоматизации процесса распределения электрических и тепловых нагрузок на ТЭЦ // Материалы XX Юбилейной междунар. конф. по вычислительной механике и современным прикладным программным системам (ВМСППС'2017, Россия, Алушта: МАИ, 2017. С. 108–110.

44. Prediction of the effective stress-strain curves of ductile polymer 1D-reinforced composites filled with hollow fibers using parameterized model based on Bezier curves / A. P. Sokolov, V. N. Schetinin, A. S. Sapelkin, M. S. Kuts, K. V. Mikhailovskiy // Key Engineering Materials. 2020. Vol. 833. P. 93–100.

45. Sokolov A. P., Schetinin V. N., Kozlov M. Yu. Surface finite element for imperfect interface modeling in elastic properties homogenization // Key Engineering Materials. 2020. Vol. 833. P. 101–106.

A. P. Sokolov, A. Yu. Pershin
Saint Petersburg Electrotechnical University

THE SYSTEM OF COMPUTER-AIDED DESIGN OF COMPOSITE MATERIALS. PART 1: PRINCIPALS, ARCHITECTURE, SOFTWARE FRAMEWORK

This paper is the first in a series of three papers on the topic of development of computer-aided design system of composite materials. The first part is devoted to the basic concepts, architectural features of the system and the description of the software framework which was developed; the second part will present the computing subsystem and the principles of organizing distributed computing using a graph-based software engineering approach; the third part is devoted to the use of a graph-based approach for developing user interaction tools with the system. This paper provides a wide overview of existing technologies and software used to automate the solution of problems that arise during the design of composite materials. The relevance of the development of domestic software in the form of an integrated system was proven. The purpose and technical requirements for the system and its architecture were defined. The paper presents a formalized approach of the development process based on the use of a component-oriented multi-tier client-server architecture. Such approaches are widely used to create industrial distributed software systems. Presented the classification and naming standards of components of the system. The concept of functional component of the system was introduced. In order to improve the performance of development of new system components, a template-based code generation subsystem has been created and used. The principles of automatic granting of access rights to functional components are presented. As one of the results of the work the software framework SA2 was developed and presented. The scheme of the developed system is presented, as well as the basic implemented subsystems are listed.

Technologies of development of engineering software, graph-based software engineering, development of computational libraries, computer-aided design of composite materials, computer-aided design, computer-aided engineering, extension modules development