

# ТЕХНОЛОГИЯ ИНТЕГРАЦИИ В ЛОГОС.ПРЕПОСТ МОДУЛЕЙ ПРЕПРОЦЕССИНГА ДЛЯ СЧЕТНЫХ ПРОГРАММ СТОРОННИХ РАЗРАБОТЧИКОВ

*Е. Н. Киселев, К. В. Иванов, А. В. Гордеев, Н. А. Мустаева, Н. И. Колпаков*

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

## Введение

Препоустпроцессор ЛОГОС.ПреПост [1] спроектирован как модифицируемая система, допускающая расширение функциональности. В данном докладе расширение функциональности рассматривается с точки зрения реализации в рамках ЛОГОС.ПреПост поддержки предварительной обработки (препроцессинга) данных для расчетных кодов сторонних разработчиков.

ЛОГОС.ПреПост – это программный модуль пакета программ «ЛОГОС» для автоматизированной подготовки и обработки расчетных моделей с целью моделирования задач аэродинамики, гидродинамики, теплопереноса, прочности, кинематики и ряда других задач.

ЛОГОС.ПреПост предоставляет графическую среду для решения на ЭВМ под управлением операционных систем Windows и Linux следующих задач:

- подготовка сеточных моделей;
- задание начальных и граничных условий;

- задание параметров моделирующих программ пакета ЛОГОС;
- запуск и сопровождение счета;
- графическая и числовая обработка результатов моделирования.

На рис. 1 изображена схема типового рабочего процесса подготовки и расчета задачи в ЛОГОС.ПреПост (штриховой линией выделены этапы работы, на которых осуществляется препроцессинг данных).

Разработка ЛОГОС.ПреПост началась в 2010 г. и изначально базировалась на требованиях к функциональности (исходных данных) методик расчета аэродинамики, гидродинамики, теплопереноса и прочности. Поэтому под сторонними разработчиками в рамках доклада подразумеваются разработчики тех методик, чьи требования к функциональности не учитывались при проектировании архитектуры ЛОГОС.ПреПост.

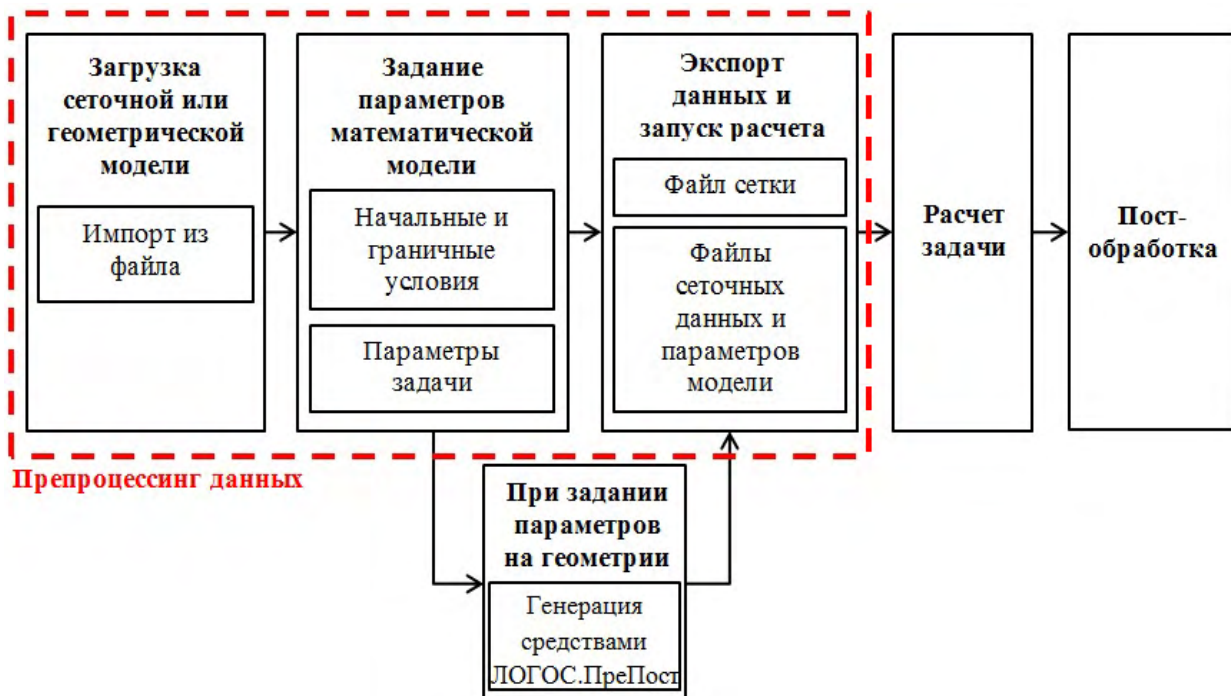


Рис. 1. Схема типового рабочего процесса подготовки и расчета задачи в ЛОГОС.ПреПост

## Описание проблемы

В рамках работ по развитию ЛОГОС.ПреПост ведется разработка графического пользовательского интерфейса (ГПИ), предназначенного для предварительной обработки данных (препроцессинга) и анализа результатов расчетов (постпроцессинга), проведенных с использованием отечественных счетных кодов.

Основным направлением развития препроцессинга является реализация средствами ЛОГОС.ПреПост дружественного интерфейса для подготовки данных, необходимых методике для проведения расчета. Проблема внедрения в ЛОГОС.ПреПост и последующего сопровождения с наименьшими трудозатратами новой функциональности (в частности, модулей препроцессинга) является актуальной, поскольку предварительная обработка данных является неотъемлемой частью процесса подготовки задач.

Модуль препроцессинга в ЛОГОС.ПреПост представляет собой совокупность программных средств, участвующих в предварительной обработке данных при подготовке задач.

В процессе интеграции модуля препроцессинга для сторонней счетной программы разработчикам ЛОГОС.ПреПост необходимо:

- обеспечить преемственность ГПИ;
- максимально задействовать имеющиеся средства формирования математических моделей ЛОГОС.ПреПост (для снижения трудозатрат на реализацию и последующее сопровождение программного кода).

Перед разработчиками методики, в свою очередь, стоит задача определения способа своего участия в процессе интеграции – ограничиться лишь предоставлением исходных данных или же, если позволяет ситуация, задействовать функциональность собственной разработки в модуле препроцессинга ЛОГОС.ПреПост.

Первый вариант, как правило, предполагает отсутствие у разработчиков методики развитого модуля препроцессинга (либо сложность его выделения из общего кода) и обладает рядом нюансов:

- возникает необходимость в подготовке разработчиками методики больших объемов исходных данных и требований к функциональной логике ЛОГОС.ПреПост;
- требуется непрерывное взаимодействие с разработчиками ЛОГОС.ПреПост (передача исходных данных, получение новых версий ЛОГОС.ПреПост, тестирование и описание выявленных несоответствий и т. д.);
- не используются наработки методики в части препроцессинга (при таком варианте взаимодействия подразумевается создание модуля препроцессинга «с нуля»);
- имеет место высокая зависимость от разработчиков ЛОГОС.ПреПост при сопровождении кода.

Главной отличительной особенностью второго варианта взаимодействия является требование к на-

личию собственной функциональности в части пре-процессинга, оформленной в виде набора динамически подключаемых библиотек. Активное взаимодействие с разработчиками ЛОГОС.ПреПост в данном случае происходит на этапе проектирования архитектуры кода и реализации программного интерфейса для подключения библиотек к ЛОГОС.ПреПост.

## Способы интеграции

В зависимости от функциональности, требуемой счетной методикой от препостпроцессора, а также исходя из имеющихся возможностей инструментария ЛОГОС.ПреПост, расширение функциональности ЛОГОС.ПреПост в части модулей препроцессинга может быть реализовано одним из следующих способов (или обоими одновременно):

- 1) редактирование текстовых файлов подсистемы конфигурирования;
- 2) добавление или замена модифицируемых компонентов, реализующих определенные программные интерфейсы.

Интеграция первым способом осуществляется преимущественно путем настройки дерева объектов графического пользовательского интерфейса (дерева модели) ЛОГОС.ПреПост и логики взаимодействия его узлов, описания хранилища и динамически генерируемых (шаблонных) диалогов через конфигурационные файлы. При этом возможна модификация или создание модулей импорта/экспорта данных, настройка визуализации и реализация нестандартного интерфейса диалоговых окон (в случае отсутствия возможности реализации требуемого «поведения» диалогового окна через реализованный в ЛОГОС.ПреПост универсальный механизм создания пользовательских интерфейсов).

Интеграция вторым способом осуществляется посредством подключения через SDK к ЛОГОС.ПреПост библиотек, реализующих логику работы дерева модели, импорт/экспорт данных, формирование объектов визуализации в интересах счетной программы. Данный способ требует модификации программных компонентов ЛОГОС.ПреПост (визуализатора, контроллера дерева модели и др.).

## Описание архитектуры модуля препроцессинга для определения набора модифицируемых компонентов

После определения способа интеграции в ЛОГОС.ПреПост новых модулей препроцессинга перед разработчиками встает вопрос определения перечня модифицируемых программных компонентов.

Основные усилия авторов доклада были направлены на систематизацию различных подходов к интеграции модулей препроцессинга для счетных программ на основе опыта внедрения различных методик большим коллективом разработчиков за не-

сколько лет развития ЛОГОС.ПреПост. Стоит все же отметить, что описанная выше классификация способов внедрения является достаточно условной, так как при интеграции одним и тем же способом модулей препроцессинга для разных методик набор модифицируемых программных компонентов ЛОГОС.ПреПост может оказаться различным.

Базовый набор компонентов ЛОГОС.ПреПост, участвующих в препроцессинге данных, схематично представлен на рис. 2.

Как видно из рисунка, в архитектуре модуля препроцессинга ЛОГОС.ПреПост можно выделить три основных уровня: уровень представления, уровень функциональной логики (бизнес-логики) и уровень данных.

Уровень представления охватывает все компоненты, которые имеют отношение к взаимодействию пользователя с программой - например, ввод управляющих команд от пользователя и вывод результатов работы на экран монитора. Сюда входит компонент визуализации, графический пользовательский интерфейс для интерактивного взаимодействия и компонент для управления деревом объектов функциональной логики (контроллер дерева модели).

Уровень бизнес-логики реализует основную функциональность системы и обеспечивает взаимодействие между данными и представлением. Для уровня бизнес-логики характерны следующие базовые понятия:

- объект – совокупность данных, представляющих собой логически выделенную единицу с точки зрения предметной области. Как правило, используется для описания интерфейса диалоговых окон (со-

держит характеристики параметров и диалоговых окон – типы данных, условия видимости, зависимости, флаги, настройки синхронизации, способы задания, свойства и т. п.);

- операция – выделенная с точки зрения предметной области законченная последовательность действий. Компоненты уровня представления могут получить доступ к объектам бизнес-логики посредством операций. Примерами операций являются добавление, удаление, модификация и чтение объектов.

Кроме объектов и операций к уровню бизнес-логики относится компонент конфигурации системы (конфигуратор), который отвечает за разбор конфигурационных файлов и трансляцию в различные структуры данных всей метаинформации о системе (описаний структуры хранилища, объектов системы, операций с объектами, моделей и другой вспомогательной информации, необходимой для работы).

Уровень данных отвечает за хранение, импорт и экспорт обрабатываемой информации. Ключевым компонентом данного уровня является хранилище данных. Хранилище данных представляет собой структуру в оперативной памяти, которая предназначена для хранения данных, описывающих текущую математическую модель – топологию сетки, наборы сеточных элементов, значения начальных и граничных условий, параметров задачи и др.

Также к уровню данных относятся импортеры и экспортеры – динамически подключаемые библиотеки для преобразования данных из различных источников в требуемый формат их физического представления.

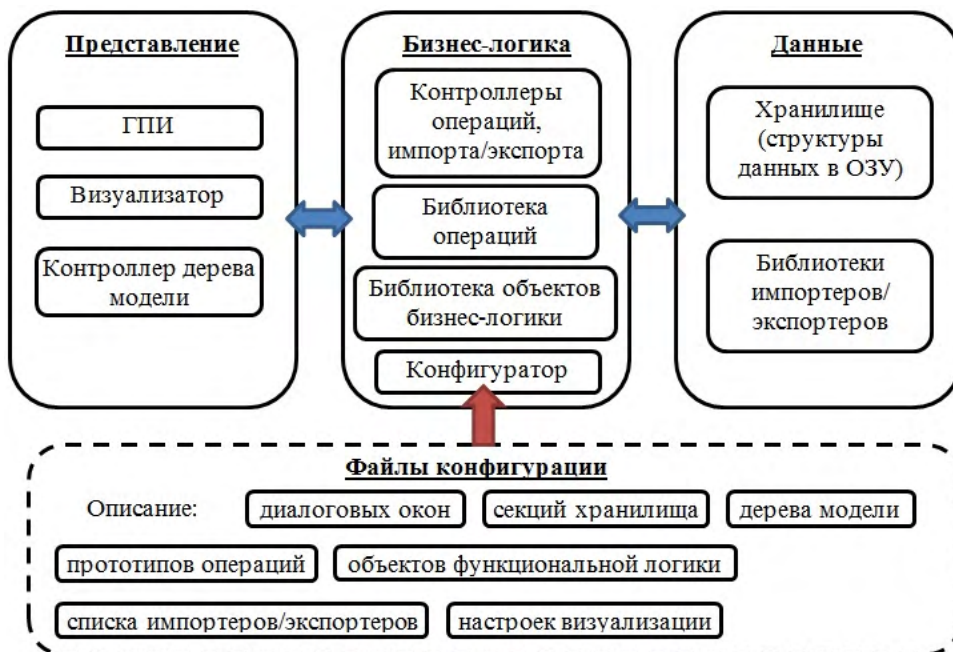


Рис. 2. Базовый набор компонентов ЛОГОС.ПреПост, участвующих в препроцессинге данных

## Описание технологии интеграции модулей препроцессинга в ЛОГОС.ПреПост

Для формирования общего представления о процессе внедрения модулей подготовки математической модели в рамках проведенной работы выполнена систематизация накопленных знаний, результат которой представлен в виде описания технологии интеграции модулей препроцессинга.

В технологическом процессе интеграции модулей предварительной обработки данных в ЛОГОС.ПреПост можно выделить следующие этапы:

- передача, анализ и обсуждение исходных данных. К исходным данным относятся:
  - общие требования к интерфейсу;
  - способы задания начальных и граничных условий (на ячейках/гранях/узлах и т.п.);
  - требования к визуализации сеток и оценка ее необходимости для целевой методики;
  - формат хранения топологии сеток (гранереберный, ячеечно-узловой);
    - тип используемых сеток (структурированные/неструктурированные);
    - размерность используемых сеток;
    - список необходимых форматов импортируемых и экспортируемых файлов;
    - необходимый инструментарий для работы с сетками (модификация, анализ и др.);
- определение способа интеграции модуля препроцессинга в ЛОГОС.ПреПост (в зависимости от готовности разработчиков методики подготовить и предоставить функциональность собственной разработки);
- определение зон ответственности разработчиков методики и ЛОГОС.ПреПост, разработка технического задания (ТЗ);
- формирование списка модифицируемых компонентов ЛОГОС.ПреПост;

- модификация программного кода компонентов ЛОГОС.ПреПост;
- тестирование и отладка реализованной функциональности.

Далее в докладе приводится описание примеров внедрения в ЛОГОС.ПреПост модулей препроцессинга для счетных программ по указанной технологии.

## Пример интеграции модуля препроцессинга в интересах методики ЛЭГАК

В данном разделе описывается внедрение в ЛОГОС.ПреПост модуля препроцессинга для методики ЛЭГАК [2] в соответствии с ранее описанной технологией. Методика ЛЭГАК служит для расчета на регулярных сетках многомерных задач газовой динамики с учетом различных физико-математических моделей детонации, а также прочности и разрушения материалов.

Интеграция выполнялась согласно техническому заданию и по совокупности требуемых изменений отнесена к первому способу интеграции, основанному преимущественно на редактировании текстовых файлов подсистемы конфигурирования.

В соответствии с выбранным способом интеграции между разработчиками были поделены зоны ответственности. Разработчики ЛЭГАК со своей стороны предоставляли программную документацию и осуществляли консультирование и передачу уточненных исходных данных, а разработчики ЛОГОС.ПреПост отвечали за модификацию программных компонентов ЛОГОС.ПреПост.

Набор модифицируемых компонентов ЛОГОС.ПреПост при внедрении модуля препроцессинга в интересах методики ЛЭГАК отмечен на рис. 3.

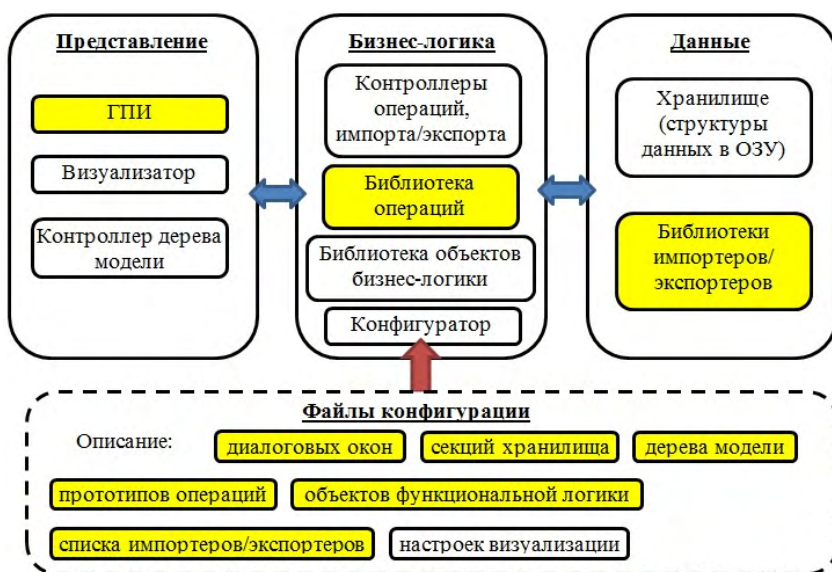


Рис. 3. Модифицируемые компоненты ЛОГОС.ПреПост при интеграции модуля препроцессинга методики ЛЭГАК (выделены цветом)

В частности, было выполнено следующее:

- добавлен большой объем текстовых описаний интерфейса и прочих настроек в файлы конфигурации;
- на уровне бизнес-логики реализован набор специфических операций для работы с данными;
- реализованы импортеры/экспортеры для работы с файлами данных в форматах ЕФР [3] и DBS [4], в которых сосредоточена и упорядочена в единую структуру вся необходимая информация о рассчитываемой задаче;
- реализован нестандартный интерфейс для ряда диалоговых окон (с табличным представлением данных и панелью управляющих элементов).

Отличительной особенностью интегрированного модуля является то, что на этапе предварительной обработки данных для методики ЛЭГАК не предполагается визуализация сеточной топологии. Кроме того, в целях экономии ОЗУ чтение массивов данных из файлов производится «на лету», т. е. непосредственно в момент обращения к ним (через соответствующее диалоговое окно).

В целом интеграция, выполненная в интересах методики ЛЭГАК, хорошо иллюстрирует ЛОГОС.ПреПост как систему, способную предоставить большой набор «штатных» инструментов для добавления нового модуля препроцессинга.

#### Пример интеграции модуля препроцессинга в интересах методики моделирования движения многокомпонентных механических систем (EULER)

Данный раздел описывает процесс внедрения модуля препроцессинга, разработанного сторонней организацией, в соответствии с ранее описанной технологией.

Интеграция модуля препроцессинга выполнялась в соответствии с техническим заданием, согласно которому требовалось реализовать программное средство (в виде библиотеки с исходными кодами, встраиваемой в ЛОГОС.ПреПост) для подготовки расчетной модели движения автомобиля при имитации дорожных испытаний.

Выбранный подход к решению поставленной задачи в общей терминологии относится к методу неавтоматизированной интеграции «на уровне пользователя» и основывается на взаимодействии разработчиков друг с другом путем обмена данными и файлами через ручное копирование, оправку почты и т. д. [5]. В рамках текущего доклада такой подход относится ко второму способу интеграции, предполагающему добавление или замену модифицируемых компонентов ЛОГОС.ПреПост, реализующих определенные программные интерфейсы.

Согласно выбранному подходу между разработчиками были поделены зоны ответственности. Разработчики сторонней организации со своей стороны готовили программную реализацию методов доступа к внутренней информации своего модуля, а разработчики ЛОГОС.ПреПост отвечали за последующее внедрение указанных методов и реализацию отображения полученных данных в ГПИ ЛОГОС.ПреПост.

Набор модифицируемых компонентов ЛОГОС.ПреПост при интеграции модуля препроцессинга в интересах методики EULER [6] отмечен на рис. 4.

Модуль препроцессинга, разработанный сторонней организацией, представляет собой набор классов и функций, участвующих в сборке программного модуля ЛОГОС.ПреПост, а также набор динамически подключаемых библиотек, которые размещаются в SDK (сокр. от software development kit – набор инструментальных средств разработки) ЛОГОС.ПреПост и предоставляют следующие программные интерфейсы для доступа к внутренней информации:

- чтение данных модели из файла \*.ELR;
- доступ к объектам модели;
- сохранение данных модели в файл \*.ELR.

Файлы \*.ELR хранят данные во внутреннем формате методики EULER в виде текстового описания параметров объектов, участвующих в процессе моделирования движения автомобиля при имитации дорожных испытаний. Фрагмент ELR-файла представлен на рис. 5.

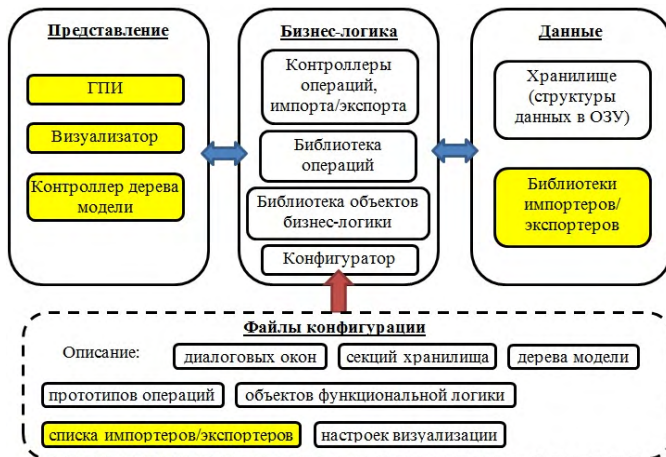


Рис. 4. Модифицируемые компоненты ЛОГОС.ПреПост при интеграции модуля препроцессинга методики EULER (выделены цветом)

```

// Простой автомобиль.
//
scalar B "колея"=1600 [ mm ];
scalar L "база"=2400 [ mm ];
scalar Mk "масса корпуса автомобиля"=1600 [ kg ];
scalar Ms1 "масса подвески колеса"=50 [ kg ];
scalar Mw1 "масса колеса"=50 [ kg ];
scalar Rs "радиус подвески колеса"=160 [ mm ];
scalar fK "частота подвески"=1 [ Hz ];
scalar ksiK "затухание подвески"=0.25;
// Данные шины
scalar B_tire "ширина профиля шины"=240 [ mm ];
scalar Rt "свободный радиус шины"=400 [ mm ];
scalar R_rim "посадочный радиус"=220 [ mm ];
scalar Rp "поперечный радиус"=400 [ mm ];
scalar FN "номинальная нагрузка"=500 [ kgf ];
scalar hN "номинальный прогиб"=20 [ mm ];
scalar Ky "коэфф. сопротивления уводу при номинальной нагрузке"=100 [ kgf / deg ];
// Параметры и условия движения
scalar V0 "начальная скорость"=30 [ km/h ];
scalar mu_s "коэффициент трения скольжения"=0.9;
//
// Кузов
point pCenter=point( -L/2, 0 [ m ], Rt );
point pK1=point( 0 [ m ], 0 [ m ], 0 [ m ] );
point pK2=point( Rt*1.5, 0 [ m ], Rt*0.5 );
point pK3=point( -Rt*1.5, 0 [ m ], Rt*1.5 );
point pK4=point( -Rt, 0 [ m ], Rt*1.8 );
point pK5=point( -Rt*2, 0 [ m ], Rt*3 );
point pK6=point( -L, 0 [ m ], 0 [ m ] );
point pK7=point( -L-Rt*1.2, 0 [ m ], Rt*0.5 );
point pK8=point( -L-Rt*1.2, 0 [ m ], Rt*1.5 );
point pK9=point( -L, 0 [ m ], Rt*3 );
line line_K1=polyLine( list( pK6, pK1, pK2, pK3, pK4, pK5, pK9, pK8, pK7, pK6 ), visible = hide );
point pK10=point( 0 [ m ], B/2-B_tire, 0 [ m ] );
point pK11=mirror( pK10, planeXZ );
solid solid_K1=extrude( line_K1, pK10, pK11 );

```

Рис. 5. Фрагмент ELR-файла в формате методики EULER

Моделирование кинематики (расчет задач по методике EULER) не требует наличия подробной сеточной топологии в виде описаний ячеек, граней и ребер – сеточная информация в ELR-файле представляет собой совокупность узлов, скаляров и полилиний, на основе которых формируются объемные геометрические тела (путем зеркалирования, экструзии и т. п.).

В соответствии со списком объектов бизнес-логики, полученных из ELR-файла, производится формирование дерева модели в ГПИ ЛОГОС.ПреПост. Пример дерева модели, сформированного в результате импорта ELR-файла, представлен на рис. 6.

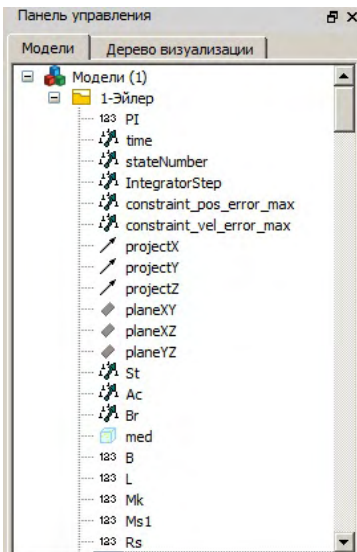


Рис. 6. Пример сформированного дерева модели «Эйлер» в ЛОГОС.ПреПост

Для формирования в окне визуализации ЛОГОС.ПреПост объемных изображений элементов конструкций на основе идеализированных сеточных элементов были реализованы программные интерфейсы для подготовки и передачи сеточных объектов в формате VTK (Visualization Toolkit – открытая кроссплатформенная библиотека трехмерного моделирования, используемая в ЛОГОС.ПреПост).

Впоследствии разработчиками ЛОГОС.ПреПост обеспечена подсветка в окне визуализации и идентификация объектов визуализации по объектам дерева модели и наоборот – переход к соответствующим узлам в дереве модели при интерактивном взаимодействии с элементами отображаемой в окне визуализации сеточной модели. Пример отображения и подсветки элементов сеточной модели представлен на рис. 7.

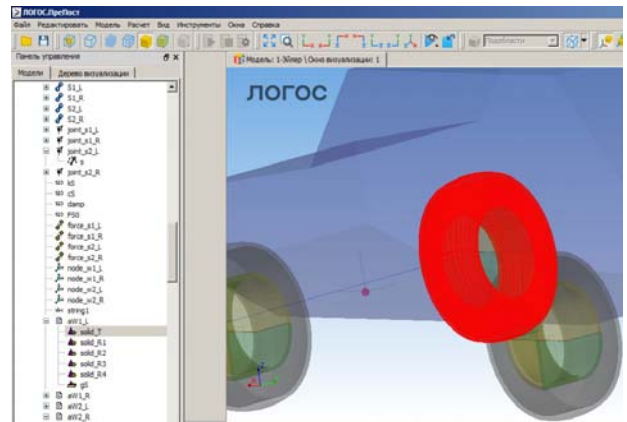


Рис. 7. Отображение и подсветка сеточных элементов модели «Эйлер» в ЛОГОС.ПреПост

## Заключение

Представленная технология внедрения модулей предварительной обработки данных для счетных методик является результатом систематизации знаний и опыта, накопленных за 9 лет развития программы ЛОГОС.ПреПост. На примере конкретных методик дано общее представление о способах интеграции модулей препроцессинга в ЛОГОС.ПреПост.

## Литература

1. Анищенко А. А., Дерюгин В. И., Дюпин В. Н., Иванов К. В., Санталов А. С., Санталова Е. Е. Препост-процессор ЛОГОС – ПРЕПОСТ, архитектура уровня бизнес – логики, хранение, импорт и экспорт данных // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2014. Вып. 2. С. 78–82.

2. Бахрах С. М., Величко С. В., Спиридонов В. Ф. и др. Методика лэгак-3d расчета трехмерных нестационарных течений многокомпонентной сплошной среды и принципы ее реализации на многопроцессорных эвм с распределенной памятью // Вопросы атомной науки и техники. Сер. Математическое моде-

лирование физических процессов. 2004. Вып. 4. С. 41–50.

3. Олесницкая К. К., Антипин И. А., Шубина М. А. Библиотека ЕФР для масштабируемого доступа к файловым данным на многопроцессорных ЭВМ / XIII Международная конференция «Супервычисления и математическое моделирование» // Сб. науч. тр. под ред. Р. М. Шагалиева. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2011. С. 101–102.

4. Артамонов М. В., Величко С. В., Дерюгин В. И., Захарова Н. А., Мустаева Н. А., Резяпов А. А. Подсистема для обработки несеточных данных по задаче и ее интеграция в комплекс ЛОГОС / XVI Международная конференция «Супервычисления и математическое моделирование» // Сб. науч. тр. под ред. Р. М. Шагалиева. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2017. С. 36–41.

5. Думченков И. А. Обзор методов интеграции информационных систем, их преимуществ и недостатков // Молодой ученый. 2018. № 23. С. 176–177.

6. Программный комплекс EULER [Электронный ресурс]: программный комплекс автоматизированного динамического анализа многокомпонентных механических систем / ООО «Автомеханика». [1993–2019]. Режим доступа: <http://www.euler.ru/index.php/euler>.