

УДК 004.514

# **Подход к автоматическому построению пользовательского интерфейса для задания параметров расчетных методик в препостпроцессоре ЛОГОС-Препост**

*В статье описан подход к автоматическому построению графического пользовательского интерфейса (ГПИ) препостпроцессора ЛОГОС-Препост. Реализованные функциональные возможности предназначены и активно используются при разработке ГПИ для задания начальных и граничных условий трехмерных задач имитационного моделирования ЛОГОС для сеточной модели. Отмечены преимущества созданного интерфейса пользователя, изложена сущность примененных решений. Предложенный подход обеспечивает гибкость, настраиваемость, быструю и легкую модификацию ГПИ приложения ЛОГОС-Препост.*

**А. А. Анищенко, В. И. Дерюгин,  
В. Н. Дюпин, А. С. Санталов**

## ***Введение***

Препостпроцессор (ППП) ЛОГОС-Препост представляет собой пакет программных модулей, предназначенных для подготовки и сопровождения счета, а также обработки результатов расчетов задач, полученных с помощью программ инженерного анализа ЛОГОС-Прочность, ЛОГОС-Аэрогидромеханика, ЛОГОС-ДАНКО, ЛОГОС-Адаптивность. Актуальной на данный момент является версия 5.3.

Одной из важнейших функций препостпроцессора ЛОГОС-Препост в рамках подготовки задачи к расчету является система задания начальных и граничных условий для сеточной модели. Для этого пользователю ППП необходим удобный и интуитивно понятный пользовательский интерфейс (ГПИ). В данной статье рассказывается о возможностях и преимуществах построения ГПИ с использованием описываемого подхода, а также о применении данного подхода для задания параметров расчетных методик.

## Начальный подход к построению ГПИ

Когда началась разработка препостпроцессора ЛОГОС-Препост, пользователи счетных методик уже имели опыт работы с различными программными пакетами, например, с коммерческими программами Ls-PrePost, Abaqus, Star-CD и другими, в том числе разработанными во ВНИИЭФ для внутреннего пользования. Соответственно, у пользователей были пожелания к облику ГПИ ППП, основанные на особенностях привычных им программных продуктов.

В начальном варианте ГПИ (версия ППП 1.0.0), который создавался на основе пожеланий пользователей, в качестве его базовых составляющих были приняты:

- дерево объектов, представляющее собой иерархию узлов; каждый из которых соответствует определенному блоку параметров счетного модуля (под блоком параметров понимается совокупность нескольких параметров); после запуска ППП в процессе работы дерево объектов практически не изменялось (рис. 1);
- набор встроенных диалоговых окон в области «мастера выполнения операций» (область в нижнем левом углу окна), активируемых одинарным щелчком мышью по элементу дерева – пожелание разработчиков модели ЛОГОС-Аэрогидромеханика (рис. 1);
- набор всплывающих диалоговых окон, активируемых двойным щелчком мыши по элементу дерева – пожелание разработчиков модели ЛОГОС-Прочность (рис. 2).

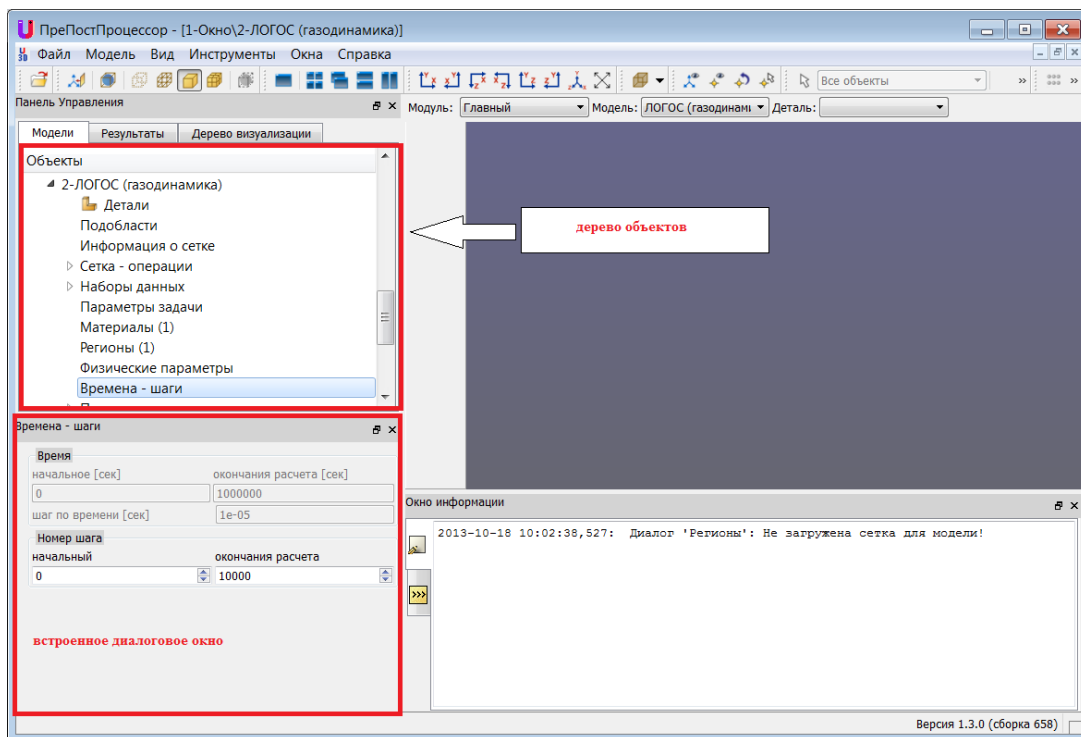


Рис. 1. Пример ГПИ с деревом объектов (в левой верхней части) и встроенным диалоговым окном «Времена – шаги» для модели ЛОГОС-Аэрогидромеханика (в левой нижней части)

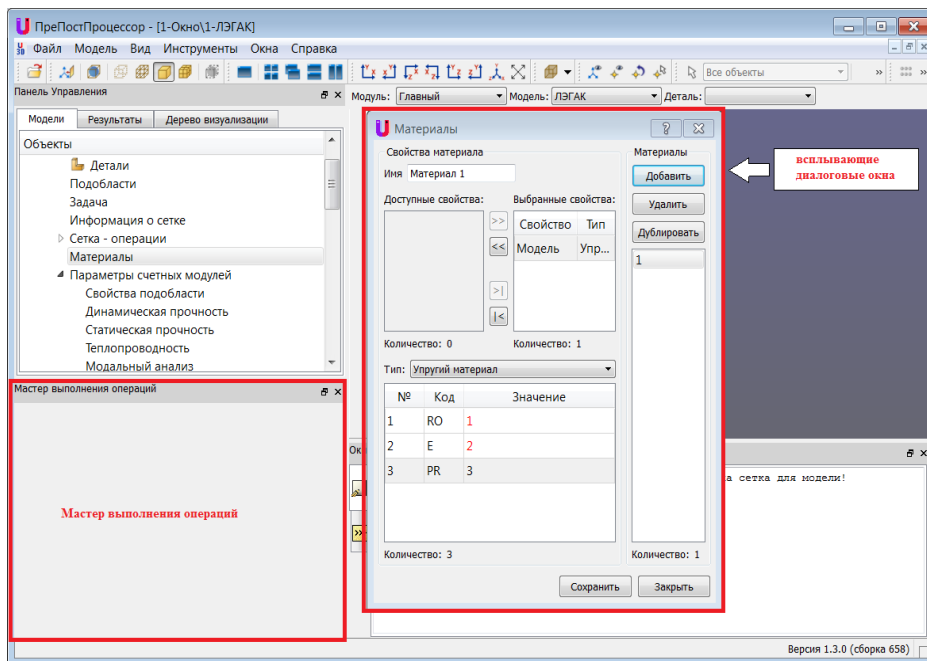


Рис. 2. Пример ГПИ с деревом объектов (в левой верхней части) и всплывающим диалоговым окном «Материалы» для модели ЛОГОС-Прочность

Недостатки этого варианта ГПИ начали проявляться сразу же после его внедрения. Коллективом разработчиков было создано большое количество (сотни) диалогов, каждый из которых нес на себе некий отпечаток видения заказчика и собственной логики разработчика. Заказчиков и разработчиков было много, поэтому все попытки обеспечить единый стиль диалоговых окон оказались неудачными. В дальнейшем стало еще труднее – заказчики стали массово просить внести изменения: добавить или изменить параметры, добавить зависимости между параметрами, изменить внешний вид диалога, сделать в некоторых случаях какие-то данные невидимыми и т. д. Причем вчера утвержденный диалог сегодня мог не понравиться, и приходилось переделывать, а иногда возвращаться к старому варианту. Стало ясно, что при таком потоке запросов разработчики просто не справятся, поэтому необходим кардинально другой подход к разработке части ГПИ, отвечающего за представление и редактирование параметров расчетных методик, основанный на использовании стандартных (шаблонных) представлений и частичной автоматизации создания диалоговых окон.

## **Формирование нового подхода к построению ГПИ**

В дальнейшем по мере накопления опыта концепция ГПИ расширялась. Для того чтобы максимально удовлетворить потребности заказчиков, в качестве основных критериев рассматривались:

- удобство в использовании;
- единый стиль и эргономика;
- минимизация трудозатрат разработчиков по изменению ГПИ при модификации и наращивании функционала системы.

Для решения поставленной задачи был использован опыт построения универсального ГПИ, полученный при разработке системы учета и контроля ядерных материалов ACCORD-2000. Архитектурные решения, примененные в ACCORD-2000, доказали свою успешность и жизнеспособность за годы ее использования на предприятиях Росатома.

Для применения имеющегося опыта к новой предметной области потребовалось также понимание тенденций развития создаваемого программного продукта и слаженная работа специалистов на всех уровнях, от разработки до использования. В результате был выработан единый и универсальный подход к построению ГПИ, предназначенного для задания параметров расчетных методик, но в то же время модифицируемого и расширяемого на более широкую область применения. Пример интерфейса ЛОГОС-Препост приведен на рис. 3.

Характерными особенностями данного подхода являются:

- представление иерархии объектов, необходимых для корректной работы расчетной методики, в виде древовидной структуры. Каждый объект, в свою очередь, может содержать в себе иерархический набор параметров. Каждый параметр может быть простым или составным (т. е. представлять собой блок параметров);
- узлами дерева могут являться блоки параметров объектов, которые можно редактировать непосредственно после активации/выбора данного узла в дереве модели;
- использование единого базового диалогового окна для редактирования параметров объектов (будем называть его шаблонным диалогом);
- возможности добавления и удаления экземпляров блоков параметров в дереве объектов с помощью контекстного меню, активируемого правой кнопкой мыши.

На рис. 3 видно, что дерево объектов отображает иерархию блоков параметров для узла «Вещества». В диалоге редактирования параметров отображаются в виде таблицы лишь параметры текущего выделенного узла; в данном случае это параметры молекулярной вязкости.

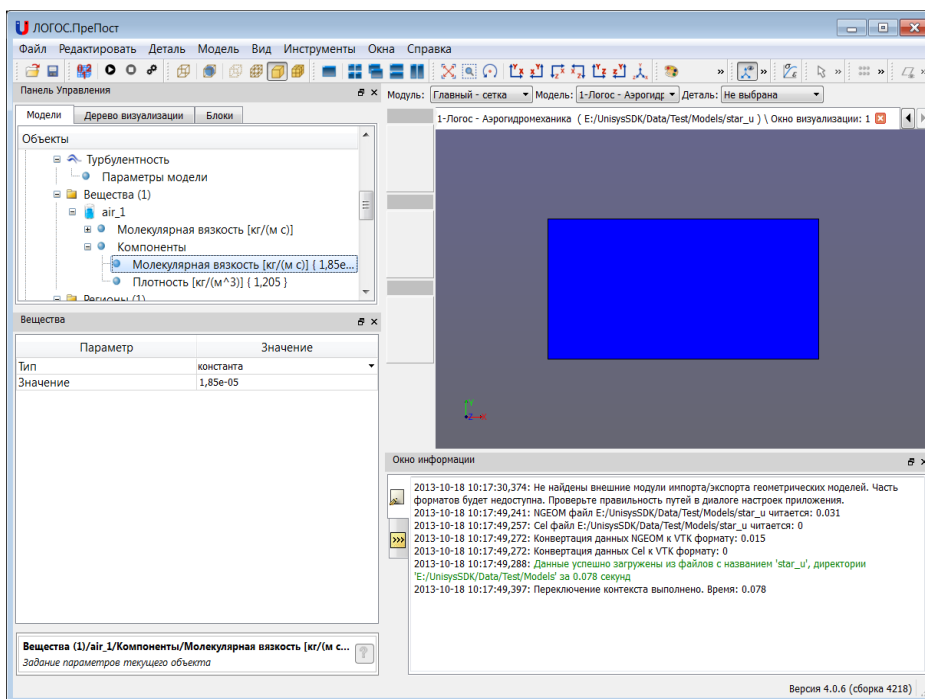


Рис. 3. Общий вид пользовательского интерфейса программного комплекса ЛОГОС-Препост

## Базовые схемы настройки и работы ГПИ

В разрабатываемой системе использовалась широко распространенная трехуровневая модель архитектуры: уровень данных, уровень бизнес-логики и уровень представления. Уровень данных отвечает за хранение и доступ к обрабатываемым данным. Уровень бизнес-логики реализует основную функциональность системы и обеспечивает взаимодействие между данными и представлением. Уровень представления охватывает все, что имеет отношение к общению пользователя с системой.

В публикациях [2] и [3] подробно рассказывается об уровне данных и уровне бизнес-логики. При этом в работе [2] описывается спецификация хранилища данных, представлен формат описания секций хранилища в конфигурационных файлах. Публикация [3] посвящена объектам и их параметрам.

Для настройки пользовательского интерфейса с использованием шаблонного диалога необходимо в конфигурационных файлах задать секцию спецификации хранилища для хранения данных нового диалога. Затем добавляется описание объекта, связанного с этой секцией хранилища. После чего объект добавляется в описание дерева объектов модели и при следующем запуске системы без перекомпиляции исходного кода ЛОГОС-Препост появится в дереве, а также будет доступен диалог для редактирования его параметров. На рис. 4 показан фрагмент конфигурационного файла для объекта «Общие параметры» модели «Аэрогидромеханика».

```
[plane] Section GENERAL_PARAMETERS
{
    enum$eUnionLogosSolverType solver_type (0);
    enum$eUnionLogosSolverTime time (0);
    enum$eUnionLogosOnOff use_GPU (0);
};
...
Object LogosUnionGeneralParameters "Общие параметры"
    DataSource "CommonData.GENERAL_PARAMETERS"
    OperationTrigger "ExecutePython_Trigger"
{
    enum$eUnionLogosSolverType solver_type "Тип решателя" ();
    enum$eUnionLogosSolverTime time "Время" ();
}
GUI
{
    helpIndex ( 2001 );
    treeObjIcon
    {
        if "${LogosUnionGeneralParameters.solver_type} == ${eUnionLogosSolverType:coupled}"
        {
            ":Icons/ico_treeIconGeneralParameters1.png"
        };
    };
    updatedTreeObjects ( "Charts;LogosUnionMaterials;LogosUnionPatches;LogosUnionPhysicalRegions" );
    contextLockMenu ( 1 );
    Parameters
    {
        "solver_type"
        {
            solver ( 1 );
        };
    };
};
...
Tree
{
    ...
    [nocount] "Общие параметры" LogosUnionGeneralParameters;
    ...
};
```

Рис. 4. Фрагмент конфигурационного файла для объекта «Общие параметры»

Ключевое слово *Section* спецификации хранилища данных относится к описанию блока данных. В спецификации описываются все необходимые секции данных со значениями параметров по умолчанию и типами данных. Поддерживаются все основные и необходимые методиками типы данных: *int*, *double*, *bool*, *enum*, *string*, *array<dimension, T>*.

В конфигурационном файле можно описывать зависимости для объекта. После ключевого слова *Object* описывается сам объект. Зависимости для объекта предполагают осуществление действий по изменению значения одного или нескольких параметров данного либо другого объекта. Такими действиями может быть изменение значения другого параметра, изменение значения параметра по умолчанию, сокращение списка возможных значений, установка дополнительных атрибутов (флагов) для параметров объекта. Механизм зависимостей позволяет также определять условия скрытия/отображения параметров объекта. Это позволяет создавать динамически формируемые графические интерфейсы.

Блок *GUI* описания объектов в конфигурационных файлах предназначен для задания специфических настроек шаблонного диалога и дерева объектов. Такими настройками могут определяться используемая пиктограмма для узла (параметра), номер раздела справочной информации, текст контекстной подсказки и др. Шаблонный диалог поддерживает работу со всеми типами данных, определяемыми спецификацией хранилища данных, и визуализирует в ходе работы ППП все происходящие изменения в соответствии с зависимостями объектов бизнес-логики (аналогичную функцию выполняет и само дерево объектов). Для задания значений параметров каждого типа используются соответствующие визуальные оконные элементы и настройки для них. Для задания параметров можно использовать динамические списки возможных значений (элементы их определяются в конфигурационных файлах), чтобы получить, например, данные другого объекта. Также разработан механизм фильтрации значений в этих списках по любым другим полям объекта. Например, в модели ЛОГОС-Аэрогидромеханика после выбора «Интерфейсы» -> «Периодические» появится шаблонный диалог, в котором при задании параметра «Набор граней» предлагается для выбора список из существующих наборов граней только с конкретным типом «CYCLIC».

Для выбора данных в элементах редактирования шаблонного диалога можно использовать пикеры – небольшие диалоговые окна для выбора значений. В качестве примера можно привести диалоговое окно выбора файла, либо пикер выбора наборов граничных условий. Пример пикера показан на рис. 5.

Наконец, существует возможность описания в конфигурационных файлах структуры собственного контекстного меню для узлов дерева модели. Можно задать в конфигурационном файле пункты контекстного меню, всплывающего по нажатию правой кнопки мыши, а также указать соответствующие обработчики, реализованные в динамических библиотеках. Можно также использовать механизм зависимостей для видимости или доступности пунктов меню.

Общая схема функционирования рассматриваемых компонентов ЛОГОС-Препост и их организация в уровни показана на рис. 6. Уровень ГПИ содержит дерево объектов, шаблонный диалог с возможностью его расширения реализациями из специально выделенных динамических библиотек (делегатов) и систему визуализации (отображения объектов), которой можно управлять посредством шаблонного диалога. Делегат представляет собой класс, содержащий набор методов, одноименных с методами шаблонного диалога и вызываемых, как правило, после вызова метода шаблонного диалога. Можно использовать нестандартные диалоги, реализуемые как на уровне исполняемого модуля приложения, так и в автоматически подключаемых на уровне бизнес-логики динамических библиотеках.

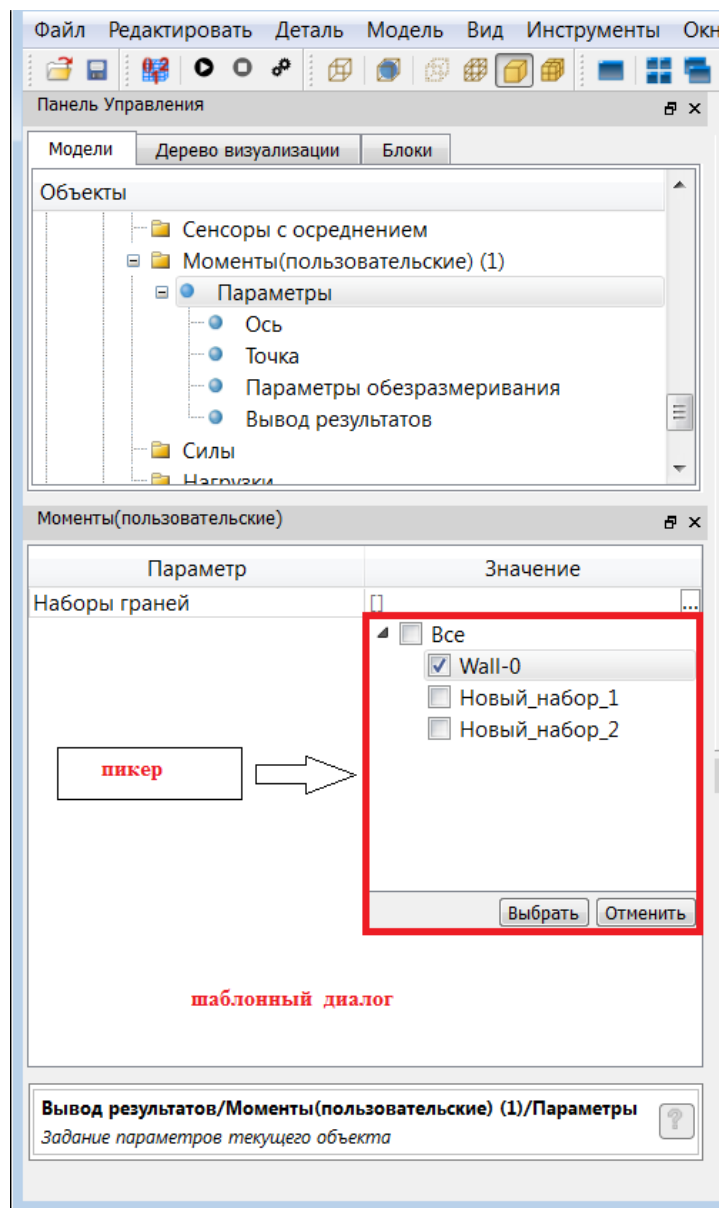


Рис. 5. Пример выбора списка параметров граничных условий в шаблонном диалоге при помощи окна пикера

Компоненты уровня бизнес-логики читают конфигурационные файлы с описанием уровня данных, уровня ГПИ и непосредственно уровня бизнес-логики. Далее информация передается на уровень ГПИ и происходит ее визуализация. Все дальнейшие обращения к уровню данных от ГПИ идут через «посредника», т. е. через уровень бизнес-логики, обеспечивающий как чтение данных, так и их запись. Это позволяет, используя средства макроязыка, протолировать операции, выполняемые пользователем, например, для повторного использования однажды пройденных сценариев. О макроязыке и операциях более подробно изложено в публикации [3].

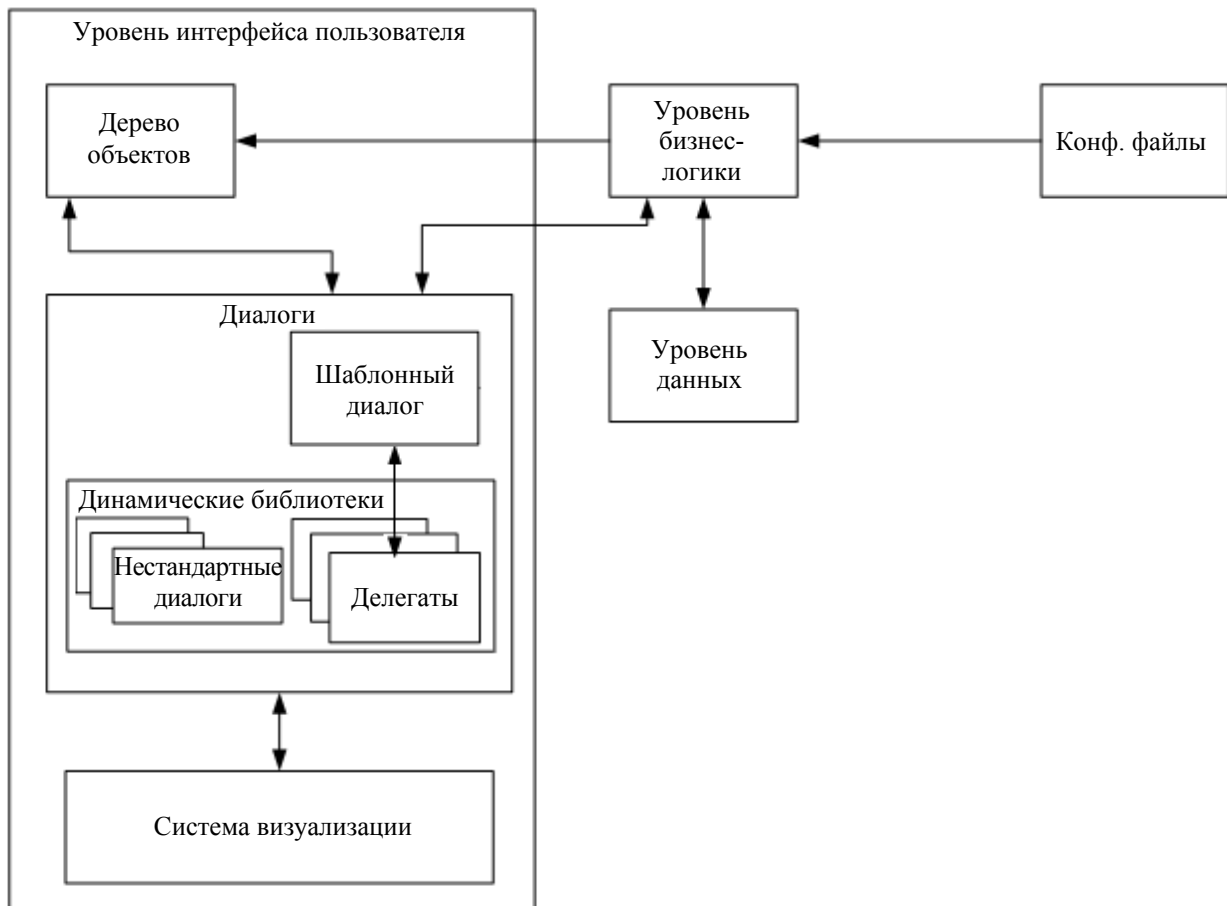


Рис. 6. Общая схема функционирования рассматриваемых компонентов ЛОГОС-Препост и их организация в уровни

### ***Текущее состояние ГПИ***

На текущий момент в стандартном представлении ГПИ используются основные возможности:

- шаблонный диалог:
  1. табличное представление и редактирование текущего блока параметров объекта бизнес-логики (поддержка пикеров, типов данных, редактирование массивов данных, элементов редактирования и т. д.);
  2. отображение элементов визуализации зависимых от параметров объектов;
  3. возможность создания делегатов диалога;
- возможность разработки любых нестандартных (например, определяемых пользователем) диалогов;
- динамически формируемое дерево объектов:
  1. добавление и удаление экземпляров блоков параметров объектов;
  2. возможность поддержки определенной структуры и функционала контекстного меню;



3. отображение и скрытие узлов дерева объектов (и пиктограмм для них) в зависимости от условий бизнес-логики;
  4. использование пикеров при обработке пунктов контекстного меню и возможность реализации функций обработчиков контекстного меню в динамических библиотеках;
- удобная клавиатурная навигация по дереву объектов и параметрам шаблонного диалога с возможностью переключения между ними при помощи комбинаций горячих клавиш;
  - синхронная работа шаблонного диалога и дерева объектов с автоматическим уведомлением друг друга о событиях модификации данных (добавление экземпляров, изменение параметров и др.), с возможностью перестроения веток дерева и обновления значений параметров диалога;
  - поддержка совместимости со старыми версиями ППП для уровня бизнес-логики.

Новый подход позволил существенно сократить трудозатраты на модификацию и сопровождение графического пользовательского интерфейса для ввода параметров расчетных методик и дал возможность в одном месте (в реализации шаблонного диалога) определять общий стиль всех диалоговых окон.

### *Заключение*

С развитием уровня бизнес-логики и выработкой стратегии по реализации единого подхода к созданию пользовательского интерфейса для задания параметров расчетных методик удалось добиться возможности развивать ГПИ на основе выбранной архитектуры с минимизацией добавления компилируемого кода на C++. Уже сегодня пользователи ППП могут модифицировать пользовательский интерфейс посредством конфигурационных файлов без участия разработчиков. Пользователи могут менять внешний вид и поведение дерева объектов и диалогов, связанных с узлами дерева, а также пиктограммы, вид и поведение контекстного меню и многое другое.

Начиная с версии 4.0.1 ЛОГОС-Препост позволяет добавлять, удалять и редактировать параметры для иерархических структур данных практически любой сложности с помощью ГПИ. На текущий момент реализованные механизмы позволяют создать пользовательский интерфейс, удовлетворяющий потребностям всех поддерживаемых методик. Например, из 475 диалогов модели ЛОГОС-Аэрогидромеханика только 6 не основаны на стандартном представлении ГПИ.

Важна и возможность быстро добавлять функционал (во многих случаях без написания кода на C++) для новых подключаемых к комплексу методик. Большая часть диалогов реализуется с использованием стандартного представления ГПИ. Использование механизма делегатов, которое требовалось на более ранних версиях, с расширением базового функционала стандартного представления практически свелось к нулю.

Авторы полагают, что дальнейшее развитие пользовательских интерфейсов для задания параметров расчетной модели будет происходить в направлении подключения других источников данных, расширения набора стандартных представлений с использованием преимуществ данного подхода, а также использования скриптовых языков (QML, QtScript, PyQt и др.) для быстрой разработки элементов ГПИ.

## **Список литературы**

1. Анищенко А. А., Бурцев С. В., Санталова Е. Е. и др. Система учета и контроля ядерных материалов Assord-2000 // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2002. Вып. 1. С. 38–46.
2. Дерюгин В. И. и др. ЛОГОС.Препост. Форматы и структуры данных // XIV Международная конференция. Супервычисления и математическое моделирование. – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2012. С. 222–225.
3. Иванов К. В., Анищенко А. А. ЛОГОС.Препост. Архитектурные решения на уровне бизнес-логики // XIV Международная конференция. Супервычисления и математическое моделирование. – Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2012. С. 329–333.
4. Дерюгин В. Н., Санталов А. С. и др. ЛОГОС-Препост. Универсальный пользовательский интерфейс задания начальных и граничных условий // Там же. С. 225–233.

## **An approach to automatically constructing the user's interface for setting parameters of computational techniques in preprocessor LOGOS-Prepost**

A. A. Anishchenko, V. I. Deryugin, V. N. Dyupin, A. S. Santalov

*An approach to automatically constructing the user's interface of preprocessor LOGOS-Prepost is described. The implemented tools are intended and actively used for the user's interface development to set the initial and boundary conditions for the grid model in LOGOS code package. The paper describes some advantages of the developed user's interface: its flexibility, adjustability, and an ability of quickly and easily modifying the interface, if necessary.*