

УДК 519.6

ПОДХОД К АВТОМАТИЧЕСКОМУ ПОСТРОЕНИЮ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ ЗАДАНИЯ ПАРАМЕТРОВ РАСЧЕТНЫХ МЕТОДИК В ПРЕПОСТПРОЦЕССОРЕ ЛОГОС-ПРЕПОСТ

А. А. Анищенко, А. С. Санталов, В. Н. Дюпин, В. И. Дерюгин
(РФЯЦ-ВНИИЭФ, г. Саров)

Описан подход к автоматическому построению пользовательского интерфейса препостпроцессора ЛОГОС-Препост. Реализованные инструментальные средства предназначены и активно используются при разработке пользовательского интерфейса для задания начальных и граничных условий для сеточной модели в пакете программ ЛОГОС. Отмечены преимущества созданного интерфейса пользователя: гибкость, настраиваемость, возможность быстрой и легкой модификации.

Ключевые слова: пользовательский интерфейс, шаблонный диалог, дерево объектов, бизнес-логика, ЛОГОС-Препост, модели, параметры счетных методик.

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 ЛОГОС-Препост 4.0.6.

Одной из важнейших функций ППП ЛОГОС-Препост при подготовке задачи к расчету является задание начальных и граничных условий для сеточной модели. Для этого пользователю ППП необходим удобный и интуитивно понятный графический пользовательский интерфейс (ГПИ). В данной статье описаны возможности и преимущества построения ГПИ с использованием предлагаемого подхода, а также применение данного подхода для задания параметров счетных методик.

Подходы к построению ГПИ

На момент начала разработки ППП ЛОГОС-Препост пользователи счетных методик уже имели опыт работы с различными программ-

ными пакетами, например с коммерческими Ls-PrePost, Abaqus, Star-CD [1–3] и др., а также разработанными в РФЯЦ-ВНИИЭФ пакетами для внутреннего пользования. Соответственно были пожелания к внешнему виду ГПИ ППП, основанные на особенностях привычных пользователям программных продуктов.

В начальном варианте ГПИ (версия ППП 1.0.0), который создавался с учетом этих пожеланий, в качестве базовых составляющих были приняты:

- *дерево объектов*, представляющее собой иерархию узлов; каждый узел соответствует определенному блоку параметров счетного модуля*. После запуска ППП в процессе работы дерево объектов практически не изменялось;
- набор встроенных диалоговых окон в области *Мастера выполнения операций* (в нижнем левом углу окна), активируемых одинарным щелчком мыши по узлу дерева, — пожелание разработчиков модели ЛОГОС-Аэрогидромеханика;
- набор всплывающих диалоговых окон, активируемых двойным щелчком мыши по узлу

*Под блоком параметров понимается совокупность нескольких параметров.

дерева, — пожелание разработчиков модели ЛОГОС-Прочность (рис. 1).

Недостатки первого варианта ГПИ стали проявляться сразу после его внедрения. Коллективом разработчиков было создано большое количество (сотни) диалогов, каждый из которых имел некие "отпечатки" видения заказчика и собственной логики разработчика. Все попытки обеспечить единый стиль диалоговых окон оказались неудачными. Ситуация усугублялась непрерывающимся потоком требований заказчиков: добавить или изменить параметры, зависимости между ними, изменить внешний вид диалога, сделать определенные данные в некоторых случаях невидимыми и т. д. Поэтому стал необходимым кардинально иной подход к разработке ГПИ, отвечающего за представление и редактирование параметров расчетных методик, который должен быть основан на использовании стандартных (шаблонных) форм и частичной автоматизации создания диалоговых окон.

В дальнейшем концепция ГПИ расширилась. Для того чтобы максимально удовлетворить потребности заказчиков, в качестве основных критериев стали рассматриваться:

- удобство в использовании;
- единообразный стиль и эргономика;
- минимизация трудозатрат разработчиков по изменению ГПИ при модификации и наращивании функциональных возможностей системы.

Для выполнения поставленной задачи был использован опыт построения универсального ГПИ, полученный при разработке системы учета и контроля ядерных материалов ACCORD-2000 [4]. Архитектурные решения, реализованные в ACCORD-2000, доказали свою успешность и дееспособность за годы использования системы на предприятиях Росатома. В результате применения имеющегося опыта в новой предметной области был выработан единообразный и универ-

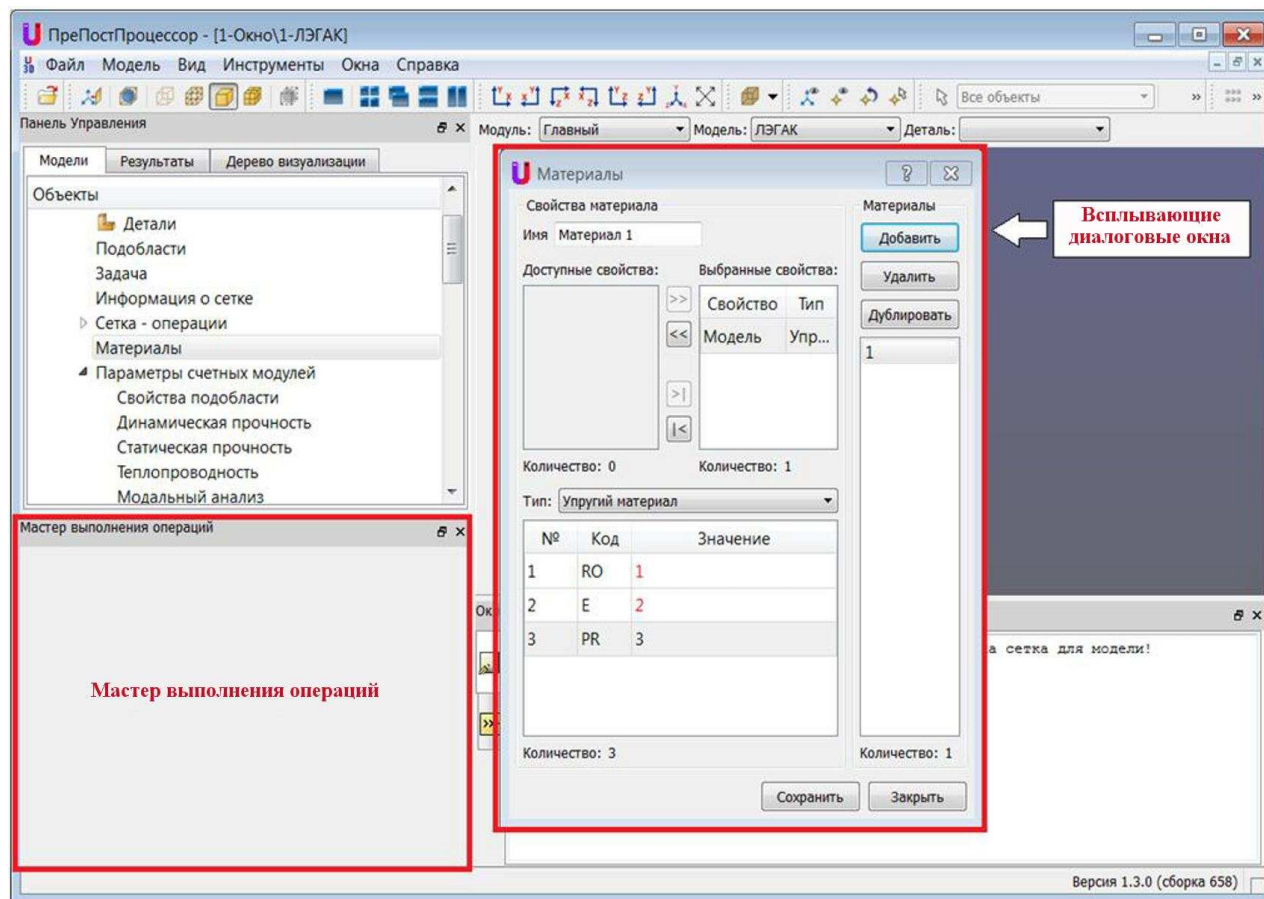


Рис. 1. Пример ГПИ с деревом объектов (слева вверху) и всплывающим диалоговым окном *Материалы* для модели ЛОГОС-Прочность

сальный подход к построению ГПИ. При таком подходе пользовательский интерфейс, предназначенный для задания параметров расчетных методик, может быть легко модифицирован и распространен на более широкую область применения.

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);
- возможность иметь в качестве узлов дерева блоки параметров объектов, которые можно редактировать непосредственно после активации/выбора соответствующего узла;

- использование единого базового диалогового окна для редактирования параметров объектов (будем называть его шаблонным диалогом);
- возможность добавления и удаления узлов блоков параметров в дереве объектов с помощью активируемого контекстного меню.

Пример нового интерфейса ЛОГОС-Препоост приведен на рис. 2. Здесь дерево объектов отображает иерархию блоков параметров узла *Вещества*. В диалоговом окне редактирования представлены в виде таблицы лишь параметры текущего выделенного узла — в данном случае это параметры молекулярной вязкости.

Базовые схемы настройки и работы ГПИ

В разрабатываемой системе использовалась широко распространенная трехуровневая модель архитектуры: уровень данных, уровень

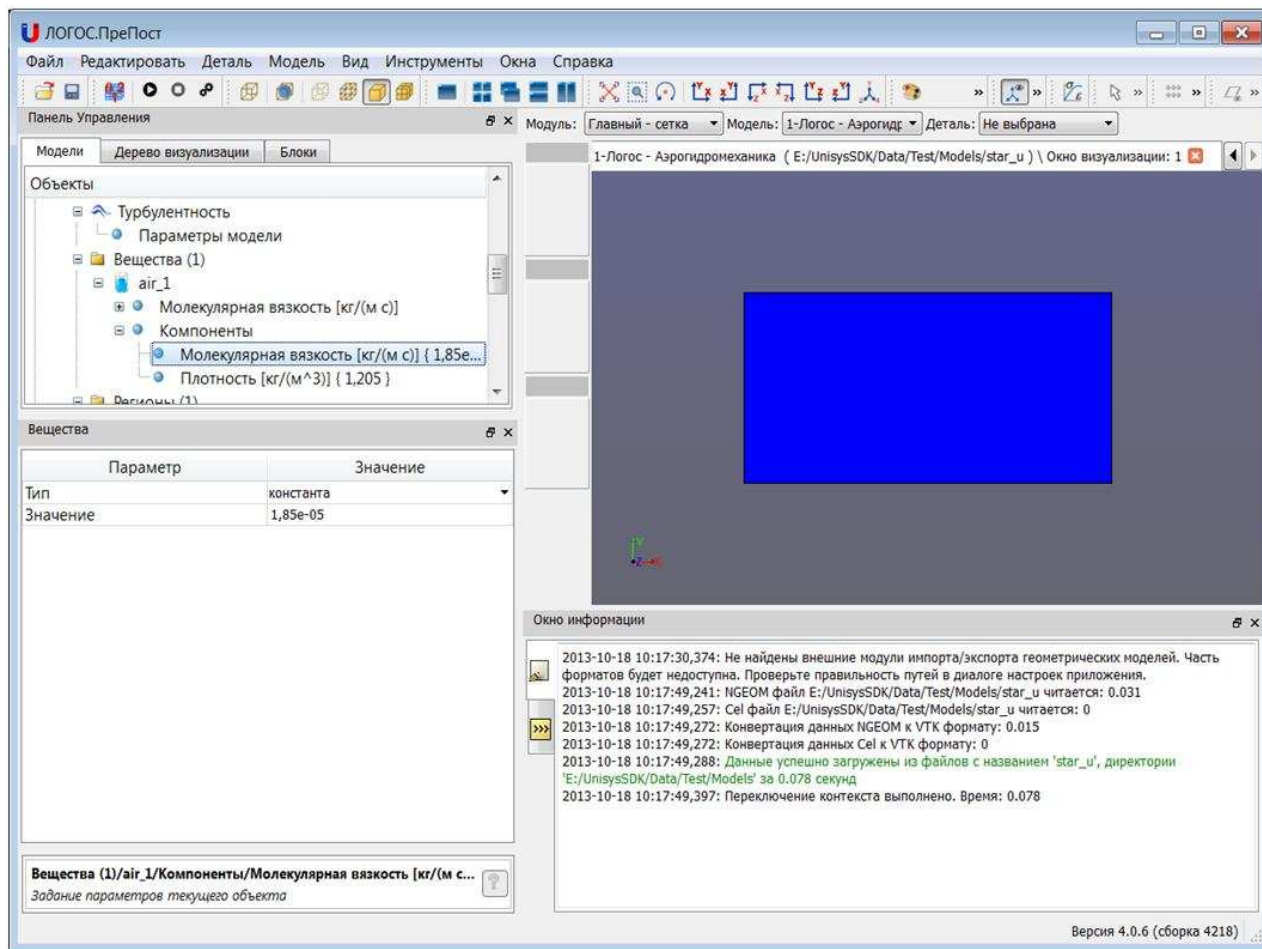


Рис. 2. Общий вид пользовательского интерфейса программного комплекса ЛОГОС-Препоост

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 публикации [5] подробно описаны уровень данных и уровень бизнес-логики: представлены спецификация хранилища данных, формат описания секций хранилища в конфигурационных файлах, объекты и их параметры.

Для настройки пользовательского интерфейса с применением шаблонного диалога необходимо в конфигурационных файлах задать секцию хранилища для данных нового диалога. Затем добавляется описание объекта, связанного с этой секцией хранилища, после чего объект добавляется в описание дерева объектов модели. В результате при следующем запуске системы без перекompиляции исходного кода ЛОГОС-Препост

объект появится в дереве, а также будет доступен диалог для редактирования его параметров.

На рис. 3 показан фрагмент конфигурационного файла для объекта *Общие параметры* модели ЛОГОС-Аэрогидромеханика.

В конфигурационном файле 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 методикам 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 выполнение действий по изменению значения одного или нескольких параметров данного или любого другого объекта. Такими действиями могут быть изменение значения одного параметра в зависимости от

```
[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$eUnionLogosSolverTime time "Время" ();
}
GUI
{
    helpIndex(2001);
    treeObjIcon
    (
        ":icons/ico_treeIconGeneralParameters1.png" : if "${LogosUnionGeneralParameters.solver_type} == ${eLogosUnionSolverType:coupled}";
        ":icons/ico_treeIconGeneralParameters2.png" : if "${LogosUnionGeneralParameters.solver_type} == ${eLogosUnionSolverType:segregated}";
        ":icons/ico_treeIconGeneralParameters3.png" : if "${LogosUnionGeneralParameters.solver_type} == ${eLogosUnionSolverType:heat}";
    );
    updatedTreeObjects( "Charts;LogosUnionMaterials;LogosUnionPhysicalRegions" );
    contextLockMenu ( 1 );
    Parameters
    {
        "solver_type"
        {
            solver( 1 );
        };
    };
};
...
Tree
{
    ...
    [pcount] "Общие параметры" LogosUnionGeneralParameters;
    ...
};
```

Рис. 3. Фрагмент конфигурационного файла для объекта *Общие параметры*

другого, изменение значения параметра по умолчанию, сокращение списка возможных значений, установка дополнительных атрибутов (флагов) для параметров объекта. С помощью механизма зависимостей определяются также условия скрытия/отображения параметров объекта. Все 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, контекстная подсказка 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 дерево объектов). При задании значений параметров каждого типа используются соответствующие визуальные оконные элементы и настройки для них. Для задания параметров можно использовать динамические списки возможных значений (их элементы определяются в конфигурационных файлах), чтобы получить, например, данные другого объекта. Также разработан механизм фильтрации значений в этих списках по любым параметрам объекта. Например, в модели ЛОГОС-Аэрогидромеханика после выбора *Интерфейсы* → *Периодические* появится шаблонный диалог, в котором при задании набора граней будет предлагаться для выбора список из существующих наборов только с конкретным типом CYCLIC.

Шаблонный диалог позволяет редактировать параметры с помощью различных компонентов ГПИ, называемых элементами редактирования (widgets): строки редактирования, выпадающего списка, флажка выбора и т. п. Для выбора нужных значений в элементах редактирования можно использовать небольшие диалоговые окна — пикеры. Примером пикера может служить диалоговое окно выбора файла. На рис. 4 приведен пикер выбора наборов граничных условий.

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

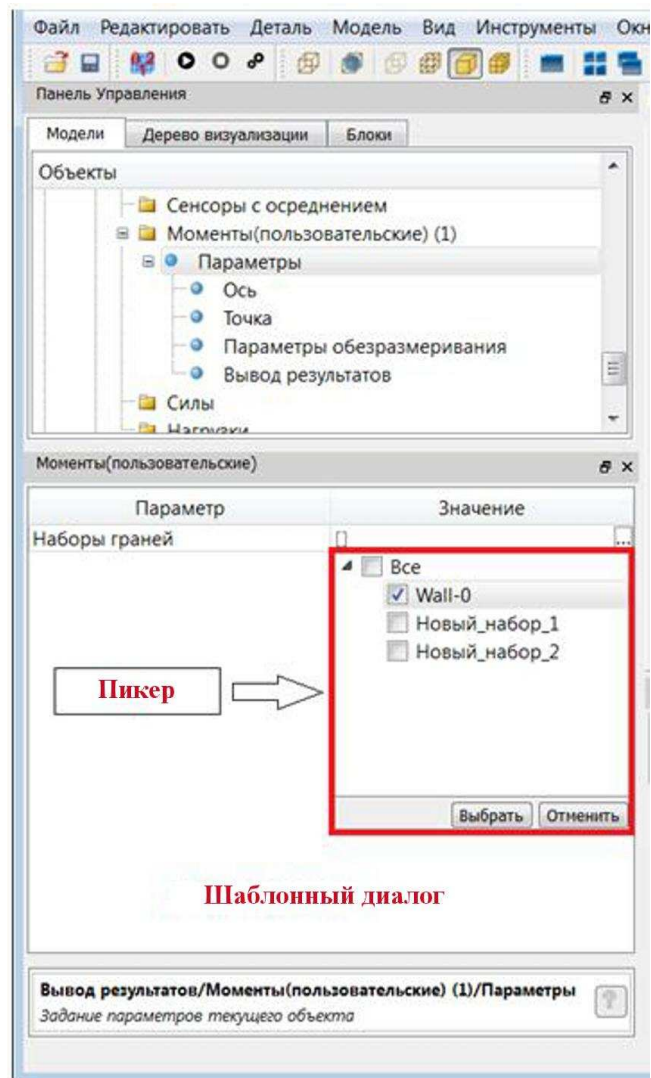


Рис. 4. Пример выбора списка параметров граничных условий в шаблонном диалоге при помощи пикера

соответствующие функции-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 на уровнях показана на рис. 5.

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 шаб-

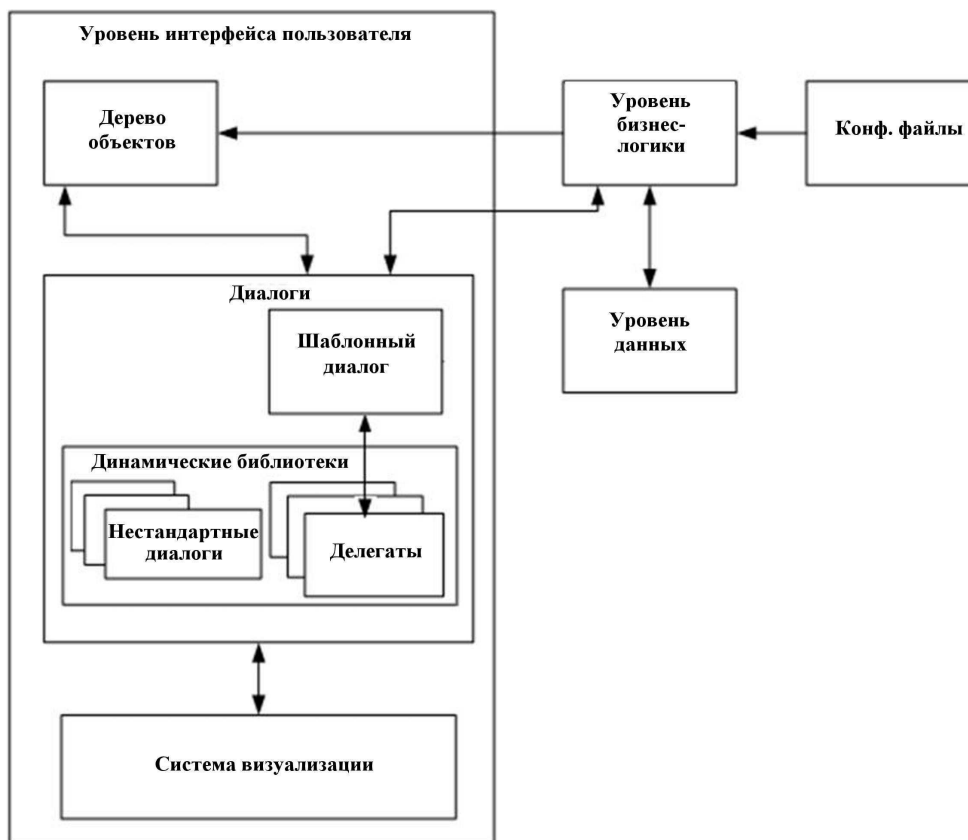


Рис. 5. Общая схема функционирования компонентов ЛОГОС-Препост и их организация на уровнях

лонного диалога и вызываемых, как правило, после вызова метода шаблонного диалога. Можно использовать нестандартные диалоги, реализуемые как в исполняемом модуле ППП, так и в автоматически подключаемых на уровне бизнес-логики динамических библиотеках.

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 осуществляются через "посредника", т. е. уровень бизнес-логики, обеспечивающий как чтение данных, так и их запись. Это позволяет, используя средства макроязыка [5], протоколировать операции, выполняемые пользователем, например, для повторного использования однажды пройденных сценариев.

Текущее состояние ГПИ

На данный момент в стандартном представлении ГПИ используются следующие основные возможности:

- шаблонный диалог:
 - табличное представление и редактирование текущего блока параметров объекта бизнес-логики (поддержка пикеров, типов данных, элементов редактирования; редактирование массивов данных и т. д.);
 - отображение элементов визуализации, зависящих от параметров объектов;
 - возможность создания делегатов диалога;
- разработка любых нестандартных (например, определяемых пользователем) диалогов;
- динамически формируемое дерево объектов:
 - добавление и удаление узлов блоков параметров;
 - возможность поддержки определенной структуры и функциональности контекстного меню;
 - отображение и скрытие узлов дерева объектов (и пиктограмм для них) в зависимости от условий бизнес-логики;
 - 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 (добавлении узлов, изменении параметров и др.), возможностью перестроения ветвей 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 ГПИ для ввода параметров расчетных методик и дал возможность определять общий стиль всех диалоговых окон в одном месте программного кода (в реализации шаблонного диалога).

Заключение

Развитие уровня бизнес-логики и выработка стратегии по реализации единого подхода к созданию пользовательского интерфейса для задания параметров расчетных методик обеспечили возможность построения 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: менять внешний вид и поведение дерева объектов и диалогов, связанных с узлами дерева, пиктограммы, вид и поведение контекстного меню и многое другое. Начиная с версии 4.0.1 ЛОГОС-Препост добавление, удаление и редактирование параметров для иерархических структур данных практически любой сложности выполняется с помощью ГПИ. На текущий момент реализованные механизмы позволяют создать пользовательский интерфейс, удо-

влетворяющий потребностям всех поддерживаемых методик. Например, из 475 диалогов модели ЛОГОС-Аэрогидромеханика только 6 не основаны на стандартном представлении ГПИ.

Важной является возможность быстрого добавления функциональности (во многих случаях без написания кода на C++) для новых подключаемых к комплексу методик с использованием стандартного представления ГПИ. Использование механизма делегатов, которое требовалось на более ранних версиях, с расширением базовой функциональности стандартного представления практически свелось к нулю.

Авторы полагают, что дальнейшее развитие пользовательского интерфейса для задания параметров расчетной модели будет происходить в направлениях 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. LS-Prepost. <http://www.lstc.com/>.
2. Abaqus Overview — Dassault Systems. <http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview>.
3. Star CCM+, Star CD. <http://www.cd-adapco.com/>.
4. Анищенко А. А., Бурцев С. В., Санталова Е. Е. и др. Система учета и контроля ядерных материалов ACCORD-2000 // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2002. Вып. 1. С. 38–46.
5. Анищенко А. А., Дерюгин В. И., Дюпин В. Н. и др. Препостпроцессор ЛОГОС-Препост. Архитектура уровня бизнес-логики, хранение, импорт и экспорт данных // Там же. 2014. Вып. 2. С. 78–82.

Статья поступила в редакцию 28.11.13.
