



УДК 004.5

© 2005 г. **В.В. Грибова**, канд. техн. наук,
А.В. Тарасов

(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

МОДЕЛЬ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ «ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС»¹

В работе представлена модель предметной области «графический пользовательский интерфейс», на основе которой формируется визуальное представление интерфейса в онтолого-ориентированном подходе для его разработки. Описаны структура модели онтологии и связи между ее компонентами.

Введение

Активное развитие программного обеспечения предъявляет к его разработке следующие основные требования: снижение стоимости разработки и особенно, упрощение сопровождения, при том, что, по оценкам специалистов, стоимость сопровождения превышает стоимость разработки в 3-4 раза. Указанные требования в полной мере относятся и к разработке пользовательского интерфейса. Более того, к пользовательскому интерфейсу предъявляются дополнительные требования: наличие средств адаптивности и настройки интерфейса для разных категорий пользователей.

Современные средства для разработки пользовательского интерфейса – Построители интерфейса (Interface Builders), Системы управления пользовательским интерфейсом (User Interface Management Systems), Моделеориентированные средства для разработки интерфейса (Model-Based Interface Development Environment) – не удовлетворяют в полной мере указанным выше требованиям. Для решения данной проблемы предлагается новый подход к разработке интерфейса, основанный на онтологиях. Учитывая указанные выше требования, предложен новый подход к разработке пользовательского интерфейса на основе онтологий [1].

Основная идея подхода – формировать декларативную модель пользовательского интерфейса на основе универсальных моделей онтологий и затем по высокоуровневому декларативному описанию автоматически ге-

¹ Работа выполнена при финансовой поддержке ДВО РАН (гранты 05-1-01-027, 05-01-01-119).

нерировать исполнимый код интерфейса. Для упрощения формирования компонента модели интерфейса – модели выразительных средств – разработана модель онтологии «графический пользовательский интерфейс», которая описывает графические интерфейсные элементы, их свойства и связь друг с другом для формирования диалога с пользователем, основанном на экранных формах.

Целью данной работы является описание модели онтологии «графический пользовательский интерфейс».

Основная концепция разработки интерфейса на основе онтологий

Для реализации основных требований современного этапа к разработке программных средств – снижения стоимости разработки и упрощения сопровождения (модифицирования) – предложен подход к разработке интерфейса на основе онтологий. Основными положениями концепции разработки интерфейса на основе онтологий являются следующие: объединение однородной по содержанию информации в компоненты модели интерфейса, формирование информации для каждого компонента модели на основе универсальной модели онтологии; автоматическая генерация кода интерфейса по этой информации.

Однородная по содержанию информация, согласно данной концепции, объединяется в компоненты модели интерфейса. При этом модель пользовательского интерфейса должна содержать всю информацию об этом пользовательском интерфейсе, которая может подвергнуться изменению в его жизненном цикле, а также допускать автоматическое построение пользовательского интерфейса по его модели (средствами компиляции или интерпретации).

Выделяются следующие составляющие модели интерфейса:

модель предметной области, в терминах которой ведется диалог пользователя с прикладной программой, в этой системе понятий выражаются входные и выходные данные прикладной программы, а также информация об интеллектуальной поддержке действий пользователя;

модель выразительных средств – визуальная составляющая интерфейса, которая обеспечивает поддержку диалога, основанного на экранных формах;

модель прикладной программы, определяющая переменные, типы их значений, общие для интерфейса и прикладной программы, а также протоколы, с помощью которых происходит коммуникация, адреса серверов, с которыми проводятся соединения, методы передачи сообщений;

модель сценария диалога, определяющая абстрактные термины для описания реакций на события (наборы действий, выполняемых при возникновении событий, источники событий, вид режимов переходов между окнами, способы выбора экземпляров окна и др.);

модель связи, определяющая соответствия между составляющими модели интерфейса.

При разработке пользовательского интерфейса необходимо, чтобы информация, представляющая входные и выходные данные прикладной программы, представлялась пользователю в виде, принятом в предметной области, для которой разрабатывается программная система и в соответствии с его требованиями. Согласно предложенной концепции для формирования каждого компонента модели интерфейса, по которой затем генерируется исполнимый код, предлагается модель онтологии, управляющая разработкой соответствующего компонента модели интерфейса. Так, разработчик интерфейса формирует информацию о предметной области – вербальное описание терминов предметной области, их свойства, связи с другими терминами. Эта информация о предметной области может быть представлена пользователю в интерфейсе различными способами (в зависимости от требований к представлению информации, типов пользователей, их предпочтений). За представление предметной области в интерфейсе «отвечает» онтология «графического пользовательского интерфейса» (ОГПИ), на основе которой формируется визуальное представление интерфейса. Эта онтология описывает состав графических интерфейсных элементов, их свойства и связь друг с другом. Графические управляющие элементы позволяют выбирать, устанавливать значения, запускать выполнение операций, а также перемещаться внутри программы. Процесс проектирования вербально представленной информации состоит в сопоставлении фрагментов предметной области с соответствующими интерфейсными элементами и конкретизации свойств интерфейсных элементов (размер, цвет, расположение на экране и др).

Модель ОГПИ

Современные графические пользовательские интерфейсы (ГПИ) являются объектно-ориентированными [1], поэтому в описании будут использоваться термины объектно-ориентированной парадигмы разработки программного обеспечения [2] и язык UML [3].

ГПИ состоят из объектов, принадлежащих различным классам. Классы могут находиться между собой в различных отношениях. В описании используются отношения обобщения (наследования) и агрегации (целое – часть).

Среди различных видов визуальных средств пользовательского интерфейса выделяются две основные группы – окна и оконные элементы управления, и три дополнительные – панели управления, оконные меню и вспомогательные средства.

Окна являются главными объектами в пользовательском интерфейсе. Они структурируют пользовательский интерфейс. Ввод-вывод данных для

каждой отдельной подзадачи осуществляется в соответствующем окне. Все остальные объекты интерфейса могут существовать только в качестве составной части какого-либо окна.

Оконные элементы управления располагаются внутри окна. С их помощью пользователь и осуществляет непосредственное взаимодействие с программой. Примеры оконных элементов: кнопки, поля ввода, статический текст, списки и т.д.

Панели управления представляют собой области, содержащие интерфейсные элементы, – как правило, кнопки, которые прикрепляются к одной из внутренних сторон окна либо помещаются поверх него. Панели управления служат для быстрого вызова команд программы. В большинстве интерфейсов по функциям они дублируют другие, более мощные, но в то же время более «медленные» средства взаимодействия с пользователем. На панелях управления размещаются средства доступа к наиболее часто применяемым командам.

Оконные меню располагаются под заголовком окна и представляют собой древовидные иерархические структуры, состоящие из пунктов меню. Каждый пункт служит для доступа к определённой команде программы. Оконные меню являются более мощным средством взаимодействия с пользователем по сравнению с панелями управления, однако для того, чтобы вызвать команду меню, в среднем требуется больше времени.

Вспомогательными средствами интерфейса называются объекты, не вошедшие в перечисленные группы, например, разделители, всплывающие подсказки и т.д.

Разделители – это элементы пользовательского интерфейса, которые располагаются внутри окна и служат для разбиения его на самостоятельные части, с каждой из которых пользователь может взаимодействовать как с самостоятельным окном. Разделитель разбивает окно на некоторое количество частей, организованных на экране в форме таблицы. Каждая ячейка таблицы может содержать собственные оконные элементы управления. Разделители могут использоваться только в многодокументном интерфейсе (см. ниже).

Всплывающей подсказкой называется прямоугольное окно со справочной информацией, появляющееся при наведении указателя мыши на интерфейсный элемент и исчезающее при перемещении указателя на другой интерфейсный элемент.

На рис. 1 приведена иерархия окон, панелей управления и оконных меню.

Выделяются три основные техники управления окнами: диалоговая, многодокументная и однодокументная и, соответственно, диалоговый интерфейс, многодокументный интерфейс (МДИ) и однодокументный интерфейс (ОДИ).

При использовании диалогового интерфейса пользователь взаимо-

действует с программой путем заполнения диалоговых форм. В этом интерфейсе используется класс Диалоговое окно.

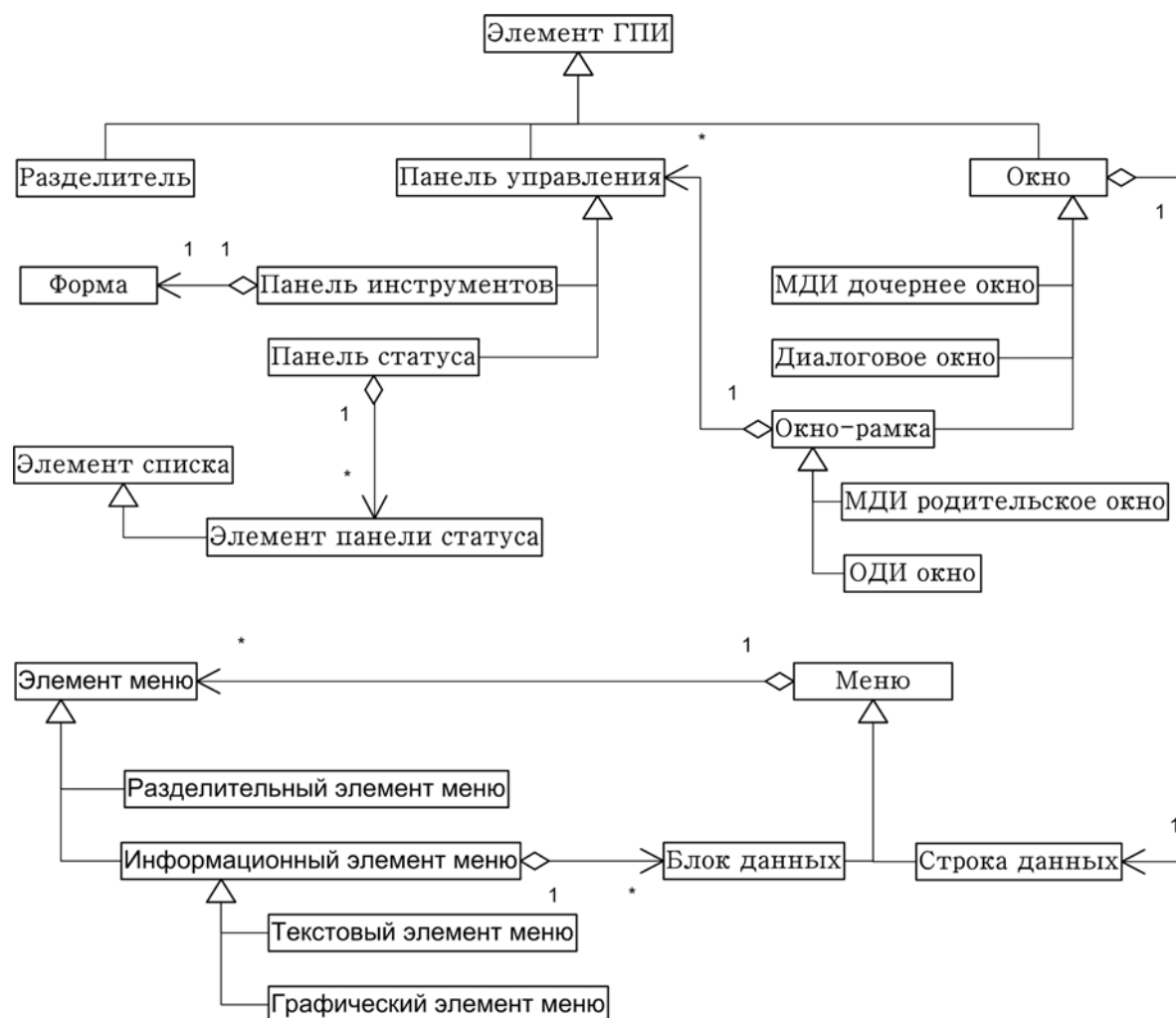


Рис. 1. Иерархия элементов ГПИ (часть 1).

При использовании МДИ пользователь взаимодействует с программой посредством так называемых «дочерних» окон, находящихся в пределах одного главного окна, осуществляющего общее управление. Такой интерфейс используется, если пользователю необходима одновременная работа с несколькими различными документами или представлениями документов. В этом интерфейсе используются классы МДИ Родительское окно и МДИ Дочернее окно.

Однодокументные интерфейсы используются в программах, в которых не нужна одновременная работа с несколькими документами. Они являются разновидностью МДИ, в которой максимальное количество дочерних окон равно одному. В этом интерфейсе используется класс ОДИ окно.

Все приведенные выше техники управления окнами могут существовать в одном и том же интерфейсе. В этом случае интерфейс называется смешанным.

На рис. 2 и 3 приведена часть иерархии элементов ГПИ, описывающая содержимое окон при МДИ и ОДИ интерфейсах.

Формой называется множество оконных элементов управления.

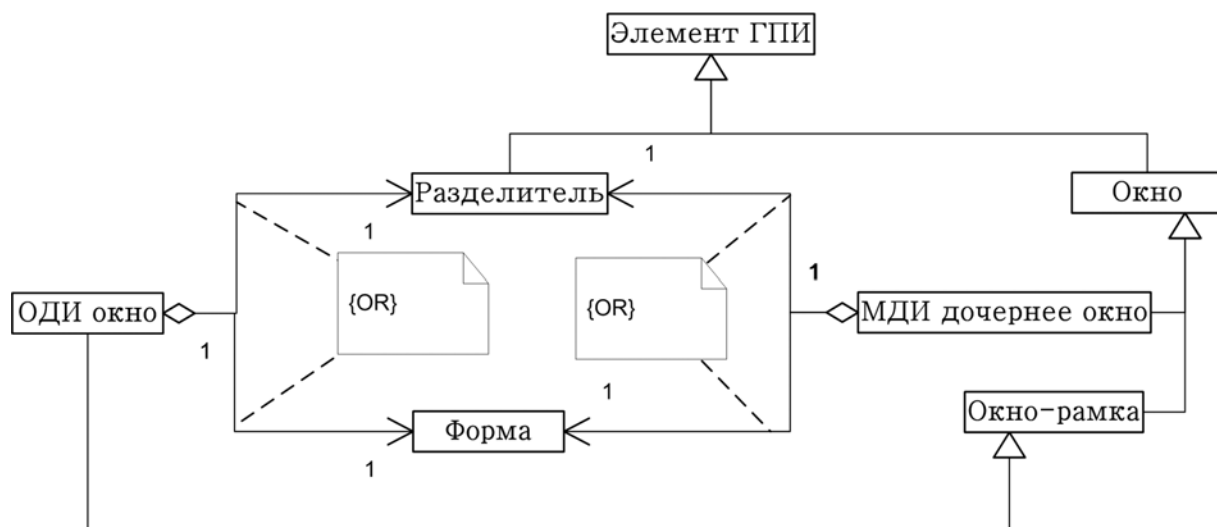


Рис. 2. Иерархия элементов ГПИ (часть 2).

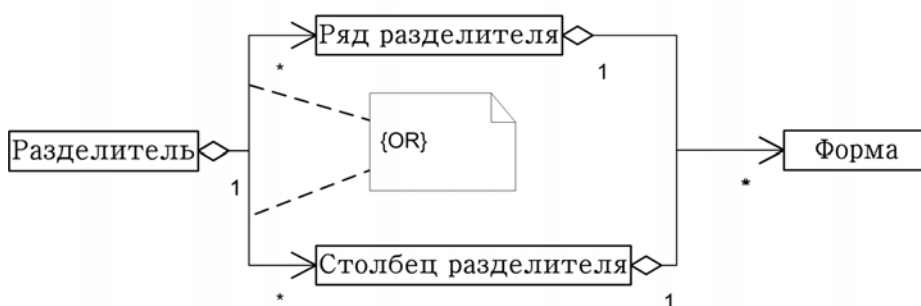


Рис. 3. Иерархия элементов ГПИ (часть 3).

МДИ «Родительское окно» и ОДИ окно могут содержать Панели управления, которые, в свою очередь, делятся на Панели инструментов и Панели статуса. Панель инструментов, как правило, располагается вверху окна и содержит элементы управления, служащие для быстрого доступа к функциям программы. Панель статуса, как правило, располагается внизу окна и состоит из ячеек, в каждой из которых отображается наиболее важная информация о состоянии программы в краткой форме.

Оконные меню подразделяются на две группы: меню в виде строки данных и меню в виде блока данных. Внутри меню в виде строки данных элементы располагаются по строкам; внутри меню в виде блока данных элементы располагаются по столбцам. Каждое меню содержит элементы, которые, в свою очередь, тоже могут содержать меню, таким образом, образуя древовидную структуру.

Существуют два вида элементов меню: разделительные и информационные. Разделительные элементы служат для группировки информационных элементов; информационные – для непосредственного вызова команд. Информационные элементы могут содержать как текстовое, так и

графическое описание.

У каждого окна может быть оконное меню в виде строки данных, содержащее команды, связанные с данным окном. Меню в виде блока данных могут присоединяться к любым интерфейсным элементам в качестве контекстного меню.

На рис. 4 изображена иерархия простых элементов управления.

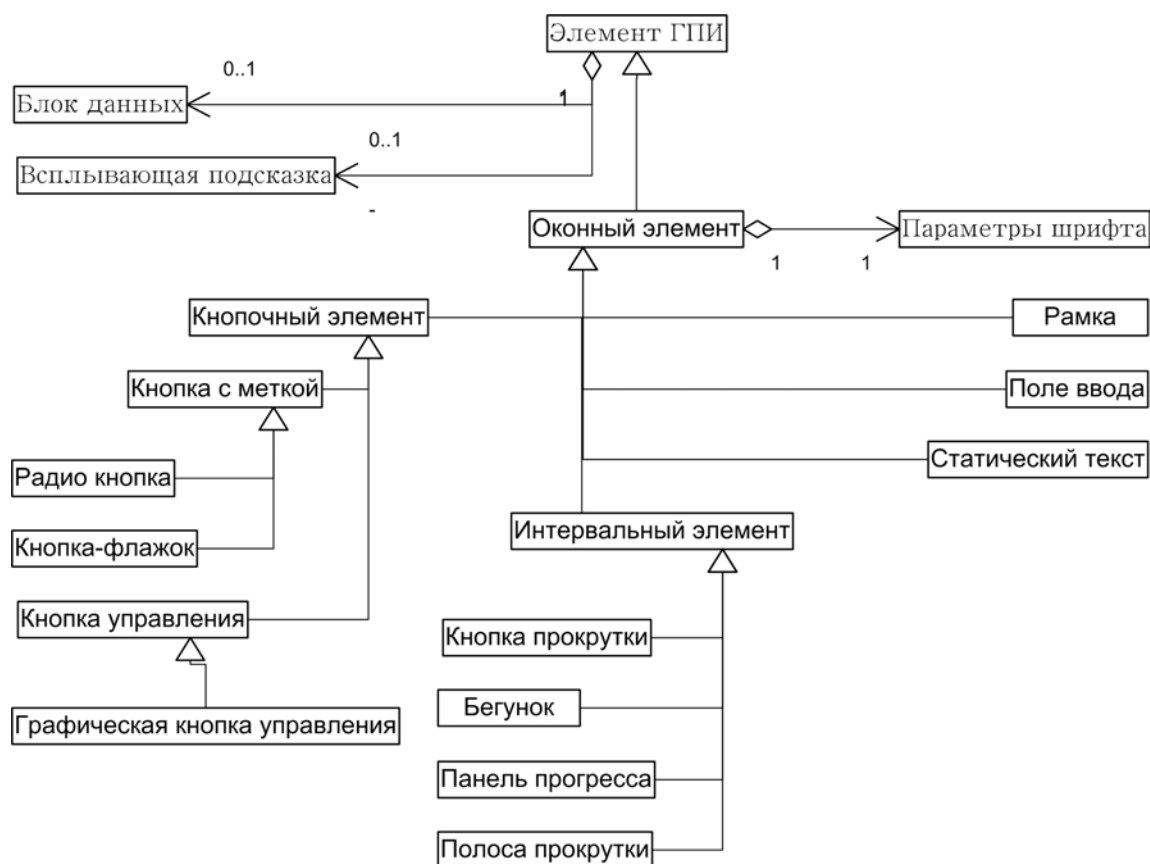


Рис. 4. Иерархия элементов ГПИ (часть 4).

В конечном итоге пользователь взаимодействует с интерфейсом посредством оконных элементов управления. У каждого такого элемента может быть всплывающая подсказка и контекстное меню. Так как каждый элемент управления может содержать текст, то у каждого элемента управления есть собственные параметры шрифта.

Кнопочные элементы управления служат для вызова команд программы либо выбора одного или нескольких вариантов из небольшого, заранее определенного множества. Интервальные элементы управления служат для представления числа из некоторого интервала. Рамка служит для группировки оконных элементов управления. Поле ввода осуществляет ввод текста. Элемент управления Статический текст служит для отображения текстовой информации.

На рис. 5 изображена иерархия составных элементов управления.

Составные элементы управления являются композицией множества однотипных частей – элементов.

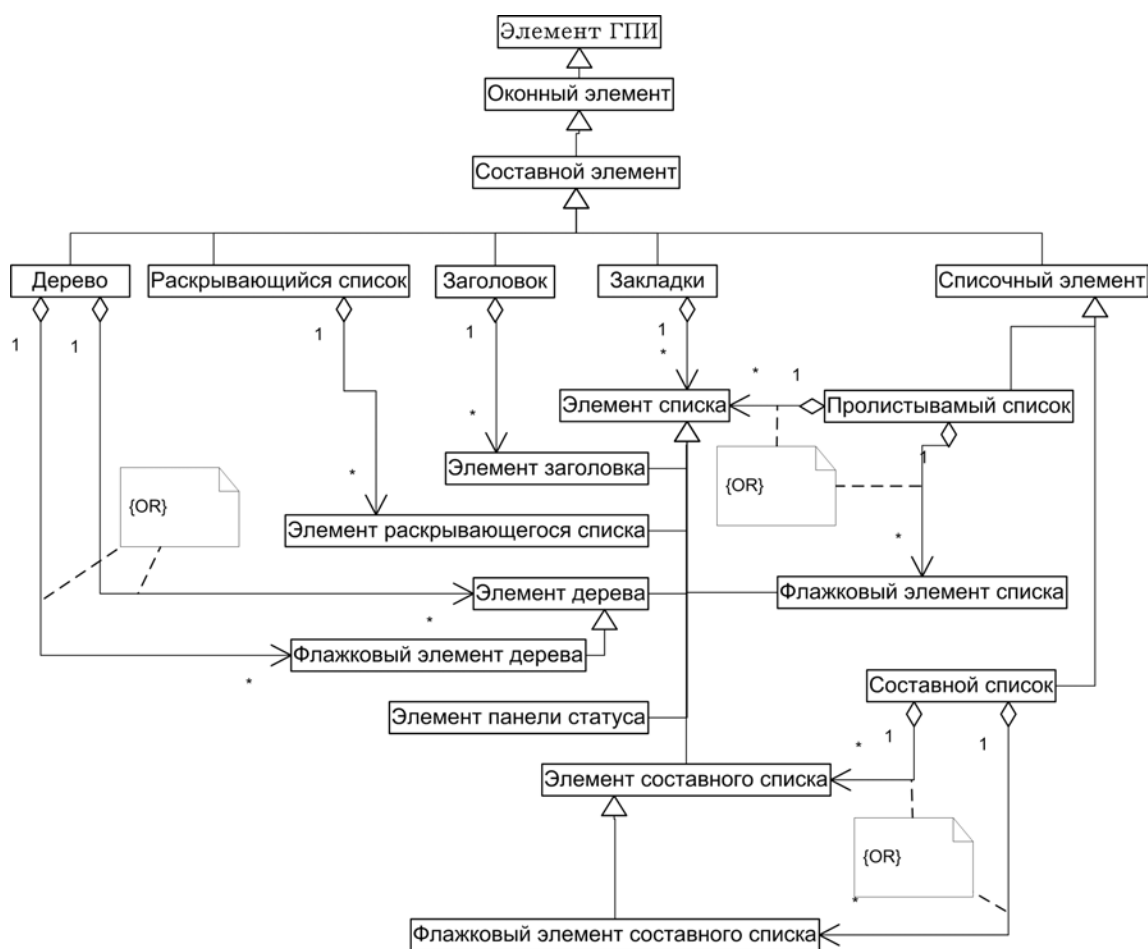


Рис. 5. Иерархия элементов ГПИ (часть 5).

Все составные элементы управления работают с текстовыми данными. Раскрывающийся список представляет собой объединение списка и поля ввода. По мере того как текст набирается в поле ввода, фокус списка перемещается в соответствии с вводимыми символами. При выборе элемента из списка текст этого элемента отображается в поле ввода. Заголовок служит для работы со столбцами данных. Закладки используются для создания логически законченных страниц или секций в пределах одного окна. Пролистываемый список представляет собой простой список строк. Составной список отличается от Пролистываемого тем, что его элементы могут быть разбиты на части. Элемент управления Дерево используется для работы с древовидными структурами.

Ниже приведены описания некоторых классов ОГПИ.

Соглашения

Классы описываются следующим образом:

<Имя класса>

Описание: <общее описание класса>

[Суперкласс: <имя класса-родителя>]

[Параметры:

{<имя параметра>:<описание параметра>. Тип: <тип параметра>}

]

Если класс А *агрегирует* класс В, то класс В называется *параметром* класса А. Класс А именуется в этом случае классом-агрегатом. Роль класса В в классе-агрегате А называется *именем параметра*.

Для описания типов параметров используются *ключевые слова* или *имена классов онтологии*. В описании типов ключевые слова (таблица) пишутся со строчной буквы, а имена классов онтологии – с заглавной буквы.

<i>Ключевое слово</i>	<i>Описание</i>
Целое	обозначает целое число
Строка	обозначает строку текста (независимо от кодировки)
Образ	обозначает некоторое графическое изображение (независимо от типа графики)
Булевский	обозначает бинарное логическое значение
Цвет	обозначает цвет (независимо от цветовой модели)
Символ	обозначает буквенный символ
Клавиатурная комбинация	обозначает клавиатурную комбинацию (независимо от конкретной клавиатуры)
Один из (список вариантов)	описывает перечислимый тип, где значением является один из вариантов
Символьная маска	описывает множество символов
Регион	описывает область на экране

Для описания *кардинальности* отношений агрегации используются *интервалы*. Пример: каждому родителю соответствует $[0, \infty]$ потомков. Для отношений «один к одному» кардинальность не указывается.

Элемент ГПИ.

Описание: класс, описывающий общие для всех элементов ГПИ свойства.

Параметры:

Х: описывает экранную координату левого верхнего угла элемента по оси Х в пикселях. Тип: целое.

У: описывает экранную координату левого верхнего угла элемента по оси У в пикселях. Тип: целое.

Координаты отсчитываются от левого верхнего угла экрана.

Ширина: описывает ширину элемента в пикселях. Тип: целое.

Высота: описывает высоту элемента в пикселях. Тип: целое.

Отображаемость: описывает, отображается ли элемент на экране. Тип: булевский.

Доступность: описывает, доступен ли элемент для взаимодействия с пользователем. Тип: булевский.

Меню: каждому элементу соответствует $[0, 1]$ контекстных меню в виде блока данных. Тип: блок данных.

Подсказка: каждому элементу соответствует $[0, 1]$ всплывающих подсказок. Тип: всплывающая подсказка.

Фон: описывает цвет фона элемента ГПИ. Тип: цвет.

Окно.

Описание: класс, описывающий свойства, области экрана, с помощью которой пользователь имеет возможность получить визуальное представление определенного аспекта решаемой задачи [4].

Суперкласс: элемент ГПИ.

Параметры.

Иконка: каждому рамочному элементу соответствует $[0, 1]$ небольших изображений, находящихся в левом верхнем углу экрана, которые используются для графической идентификации окна. Тип: образ.

Описывающие заголовок окна.

Заголовок: описывает наличие заголовка. Тип: булевский.

Текст: описывает текст заголовка. Тип: булевский.

В правой части заголовка может находиться ряд кнопок, называемых системным меню. Эти кнопки служат для управления состоянием окна.

Системное меню: описывает наличие системного меню. Тип: булевский.

Системные кнопки: каждому рамочному элементу соответствует $[0, \infty]$ системных кнопок. Тип: образ.

Существуют два слоя окон: верхний и нижний. Окна нижнего слоя не могут перекрывать окна верхнего слоя. Окна верхнего слоя всегда перекрывают окна нижнего слоя.

Верхнее окно: описывает, принадлежит ли окно верхнему слою. Тип: булевский.

Окна могут быть модальными и не модальными. Модальные окна не позволяют пользователю взаимодействовать с другими окнами приложения.

Модальность: описывает, является ли окно модальным. Тип: булевский.

Описывающие наличие прокрутки:

Горизонтальная: описывает наличие горизонтальной прокрутки. Тип: булевский.

Вертикальная: описывает наличие вертикальной прокрутки. Тип: булевский.

Вертикальная слева: описывает, находится ли вертикальная полоса прокрутки в левой части поля ввода. Тип: булевский.

Состояние: описывает текущее состояние окна. Окно может быть развернуто во весь экран (максимизировано), минимизировано (удалено с экрана) и находиться в нормальном состоянии. Пользователь может изменять размеры только тех окон, которые находятся в нормальном состоянии. Тип: один из (максимизировано, минимизировано, норма).

Изменяемые размеры: описывает, может ли пользователь изменять размеры окна, если оно находится в нормальном состоянии. Тип: строка меню.

Граница: описывает наличие границы окна. Тип: булевский.

Регион: описывает отображаемую часть окна. Тип: регион.

Заключение

Модель онтологии «Графический пользовательский интерфейс» используется для формирования одного из компонентов модели интерфейса - модели выразительных средств в рамках онтологоориентированного подхода к разработке интерфейса. В настоящее время онтология графического пользовательского интерфейса, описанная на языке OIL, доступна в Интернете [5]. Представленная онтология используется также в учебном процессе на кафедре программного обеспечения ЭВМ в рамках курса «Человеко-машинный интерфейс». На основе данной модели онтологии в отделе интеллектуальных систем ИАПУ ДВО РАН разработана база знаний о дефектах интерфейса.

ЛИТЕРАТУРА

1. *Kleshchev A.A., Gribova V.V.* An ontology-oriented approach to user interface development. International Journal "Information Theories & Applications. 2003. Vol. 10. Num.1. P. 87-94.
2. *Мандел Т.* Разработка пользовательского интерфейса: Пер. с англ. М.: ДМК Пресс, 2001.
3. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд./Пер. с англ. М.: Изд-во БИНОМ; СПб.: «Невский диалект», 2000.
4. *Буч Г., Рамбо Дж., Джекобсон А.* Язык UML. Руководство пользователя. / Пер. с англ. М.: ДМК, 2000.
5. *Гультяев А.К., Машин В.А.* Проектирование и дизайн пользовательского интерфейса. СПб.: КОРОНА-принт, 2000.
6. <http://interface.es.dvo.ru>

Статья представлена к публикации членом редколлегии А.С. Клещевым.