

ОНТОЛОГИЯ ПРЕДМЕТНОЙ ОБЛАСТИ КОМПОНЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Борискин А. С., Синельников П. М., Стельмачёнок М. О.

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: {stelmachenokm, pavelsinelnikovbel}@gmail.com, coloss_000@mail.ru

В данной работе будет рассмотрена иерархия компонентов пользовательского интерфейса и пример спецификации для такого компонента

ВВЕДЕНИЕ

Отсутствие унификации в принципах построения пользовательских интерфейсов препятствует распараллеливанию процесса разработки пользовательского интерфейса (ПИ) и возможности повторного использования уже разработанных компонентов ПИ. Предлагаемый в рамках Технологии OSTIS[1] онтологический подход к проектированию пользовательских интерфейсов решает эту проблему путём использования библиотеки многократно используемых компонентов, в рамках которой специфицированы различные компоненты ПИ.

Одним из принципов онтологического подхода является то, что каждый элемент управления пользовательского интерфейса является внешним отображением некоторого элемента, хранящегося в памяти. Таким образом, описываемая онтология **предметной области компонентов пользовательских интерфейсов** позволит корректно трактовать прагматику и семантику объектов на экране, а также адресовать интеллектуальной системе запросы, касающиеся организации пользовательского интерфейса.

I. ОПИСАНИЕ ОСНОВНЫХ ПОНЯТИЙ ОНТОЛОГИИ

На верхнем уровне типология компонентов пользовательского интерфейса выглядит следующим образом:

компонент пользовательского интерфейса

\leq разбиение*:

{

- атомарный компонент ПИ
- неатомарный компонент ПИ

\Rightarrow включение*:

- Ядро ПИ
- ПИ среды проектирования
- ПИ среды визуализации и редактирования

}

Ядро пользовательских интерфейсов

на нижнем уровне состоит из модели базы знаний, включающей в себя некоторое подмноже-

ство базы знаний ПИ, и модели машины обработки знаний, состоящей из некоторого коллектива агентов, характеризующих интерфейсную деятельность ostis-системы и пользователя. На верхнем уровне же Ядро представляет собой множество компонентов ПИ различной сложности и назначения.

Быстрое прототипирование является одним из основных достоинств Технологии OSTIS, препятствующих её старению, что приводит к необходимости введения версионности на этапе построения библиотеки многократно используемых компонентов. Это значит, что новая версия Ядра ПИ ostis-системы, по сути, формируется из новых версий компонентов ПИ, включенных в состав этого Ядра. Под **компонентом ПИ** подразумевается знак фрагмента базы знаний[2], имеющий определённую форму внешнего представления на экране и являющийся аргументом для некоторого подмножества интерфейсных команд.

Установим различия в *семантической* и *прагматической составляющей* любого компонента пользовательского интерфейса. **Семантическая составляющая** заключается в определении того, знаком какой сущности является отображаемый на экране компонент. **Прагматическая составляющая** рассматривает прикладной аспект (аспект применения) отображаемого на экране компонентах[3].

На уровне sc-памяти имеет значение только семантическая составляющая, однако данный факт не влияет на процесс эксплуатации системы пользователем, поскольку обе составляющие отражают разные стороны одного и того же знака некоторой сущности. Например, за каждым окном ostis-системы скрывается локализованный sc-текст некоторого сообщения, а за каждой кнопкой – знак некоторого класса действий, инициируемого при нажатии клавиш устройства ввода.

Рассмотрим типологию компонентов ПИ:

компонент пользовательского интерфейса

\Rightarrow включение*:

- элемент управления
- контейнер

- *курсор*

Под **элементом управления** понимается знак компонента пользовательского интерфейса, с помощью которого пользователь имеет возможность повлиять на состояние как отображаемых на экране объектов, так и на состояние элементов, знаки которых и скрываются за каждым из этих объектов.

Под **контейнером** понимается знак компонента пользовательского интерфейса, задача которого состоит в размещении набора компонентов, включённых в его состав.

Наконец, **курсор** определяется как компонент пользовательского интерфейса, указывающий на компонент пользовательского интерфейса, с которым будет производиться взаимодействие.

II. СПЕЦИФИКАЦИЯ КОМПОНЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Любой компонент ПИ, входящий в состав Ядра ПИ *ostis*-системы, подлежит обязательной спецификации, для всех остальных компонентов эта процедура носит рекомендательный характер. Описание спецификации пользовательского интерфейса предполагает наличие следующих пунктов:

- **Имя компонента:** при выборе имени дается предпочтение прагматической составляющей компонента пользовательского интерфейса, поскольку она ближе для понимания обычного разработчика, на теоретико-множественном же уровне описания компонента в полной мере раскрывается семантическая составляющая;
- **Тип компонента:** в этом пункте мы показываем теоретико-множественные связи, заданные на компоненте;
- **Версионность:** наличие версий компонента ПИ позволяет в условия открытой разработки фиксировать лиц, причастных к разработке компонента, а также отслеживать изменения, произошедшие с данным компонентом ПИ на этапе всего его существования, откатываясь до предыдущих версий в условиях конфликта с другими компонентами ПИ;
- **Разработчики:** лица, имеющие отношение к разработке компонента ПИ;
- **Описание компонента:** информация о назначении компонента ПИ и о компонентах ПИ, входящих в состав рассматриваемого.

- **Граница*** - определяет форму внешнего представления компонента пользовательского интерфейса. На линейной геометрической фигуре, являющейся границей* компонента ПИ, могут быть заданы линейные размеры, например, длины сторон полученного многоугольника и т.д.
- **Координата*** - определяет положение компонента пользовательского интерфейса в пределах некоторой системы координат (экрана, в частности). В зависимости от вида границы* компонента пользовательского интерфейса, определяются различные точки отсчёта*: для прямоугольного окна это, например, координата* левого верхнего пикселя, для круглого же окна это координата центра* этого окна.
- **Цвет** - параметр, задающий вид пространства внутри границы* компонента ПИ. Использование теоретико-множественных отношений включение* и разбиение*, позволяют более интересную заливку компонента ПИ, формировать стили.

ЗАКЛЮЧЕНИЕ

В работе была рассмотрена онтология предметной области компонентов пользовательского интерфейса, позволяющая унифицировать все элементы, расположенные на экране. Использование же спецификаций облегчает процессы мониторинга этих компонентов и интеграции их с различными версиями ядра интеллектуальной системы.

III. СПИСОК ЛИТЕРАТУРЫ

1. Голенков, В. В., Гулякина, Н. А. - Семантическая технология компонентного проектирования систем, управляемых знаниями. – В книге Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2015). Материалы конф. – Минск: БГУИР, 2015.
2. Davydenko, I. T. Ontology-based knowledgebase design, / I. Davydenko // Open Semantic Technologies for Intelligent Systems: материалы междунар. науч.-техн. конф./ редкол.: В. В. Голенков (отв. ред.) [и др.], ISSN 2415-7740; Вып.1 (Минск, 16-18 февр. 2017г.). - Минск: БГУИР, 2017. С.57-72.
3. Борискин, А. С., Жуков, И. И., Корончик, Д. Н., Садовский, М. Е., Хусаинов, А. Ф. - Онтологическое проектирование пользовательских интерфейсов интеллектуальных систем. – В книге Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2017). Материалы конф. – Минск: БГУИР, 2017.