ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2019 Управление, вычислительная техника и информатика № 49

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.514.6
DOI: 10.17223/19988605/49/13

В.В. Грибова, Л.А. Федорищев

АДАПТИВНЫЙ ГЕНЕРАТОР-WIMP-ИНТЕРФЕЙСА РЕДАКТОРОВ БАЗЫ ЗНАНИЙ НА ОСНОВЕ ОНТОЛОГИИ

Работа выполнена при частичной финансовой поддержке фонда РФФИ № 17-07-00956, № 19-07-00244 и КПФИ «Дальний Восток» № 18-5-078.

Рассматриваются концепция и основные принципы проектирования, реализации и адаптивной генерации пользовательских WIMP-интерфейсов облачных редакторов баз знаний и данных по их онтологии и базе знаний о правилах проектирования интерфейсов. Для иллюстрации предложенного подхода приведены концептуальные схемы генерации интерфейса и представлены соответствующие примеры.
Ключевые слова: WIMP-интерфейсы; онтологии; адаптивность; редактор базы знаний.

Пользовательский интерфейс является неотъемлемым компонентом подавляющего большинства программного обеспечения (ПО). Его разработка достаточно трудоемка и, по оценкам различных специалистов, занимает не менее половины времени, требуемого на разработку ПО в целом [1, 2]. Разработчику интерфейса приходится учитывать требования пользователей, их опыт, предпочтения, контекст использования программного средства, знания стандартов разработки, руководство по стилю целевой платформы и т.п. Таким образом, качество пользовательского интерфейса целиком и полностью зависит от навыков и умений его разработчика. Поэтому актуальность исследований в области автоматизации разработки пользовательских интерфейсов с учетом большого количества факторов очевидна.

Среди множества классов ПО выделяется особый класс – системы с базами знаний (СБЗ), которые в настоящее время активно используются для решения многих научных и практических задач; можно говорить о периоде зрелости данного класса систем. Несмотря на ряд успехов, достигнутых как российскими так и зарубежными коллективами, разрабатывающими инструменты для их проектирования, одной из острых остается проблема формирования базы знаний экспертами предметной области без посредников в лице инженеров знаний и программистов. Одним из решений этой проблемы является использование онтологического подхода.

Существующие редакторы баз знаний, в том числе на основе онтологий – IWE, Protégé, OntoEdit, GrOWL, Graphl, RDFGravity, WebVOWL, Ontolingua, OntoSaurus, OilEd, WebOnto, WebODE и др., предлагают пользователям различные интерфейсные решения. В большинстве из них интерфейс редактора основан на экранных формах (WIMP-интерфейс[1]). В некоторых редакторах интерфейс с помощью плагинов или встроенных средств может быть также расширен другими видами представлений (графовые структуры, графические схемы, таблицы и др.), которые реализуют, как правило, только просмотр информации; ее ввод и редактирование осуществляются с помощью WIMP-интерфейса как наиболее распространенного и интуитивно понятного. Однако, несмотря на разнообразие интерфейсных решений, они имеют «жесткий» интерфейсный стиль, не адаптирующийся *к структуре содержимого, внутреннему или внешнему окружению, изменению типа пользователя*, работающего с ПО, и тому подобному в соответствии с требованиями проектирования (стандартами

[1] WIMP-интерфейс (windows, icons, menus, pointers – окна, пиктограммы, меню, указатели) в человеко-компьютерном взаимодействии означает взаимодействие с компьютером на базе этих элементов.

юзабилити[1]). В то же время важность адаптивных технологий в области разработки интерфейсов активно обсуждается в литературе [3, 4], при этом во многих приложениях (компьютерных играх, ряде мобильных сервисов) именно адаптивные интерфейсные технологии позволили вывести их на новый уровень, благодаря умению «интеллектуально» подстроиться под каждые задачу и пользователя. Учитывая, что разработчиками баз знаний являются эксперты в предметной области (в соответствии с современным подходом к разработке интеллектуальных систем), имеющие разный уровень владения компьютерными технологиями, исследования в области создания пользовательских интерфейсов, учитывающих структуру исходных данных, устройство, платформу, предпочтения и особенности пользователя, для редакторов баз знаний и сложноструктурированных данных являются актуальными.

Целью работы является описание концепции создания адаптивных генераторов пользовательских WIMP-интерфейсов по онтологии знаний. Описана архитектура программного средства и ее основные компоненты.

1. Концепция автоматизации разработки адаптивных редакторов баз знаний

В настоящее время наиболее распространенным является подход к формированию баз знаний на основе онтологий. В работах [5–13] подробно обсуждаются преимущества такого подхода. В соответствии с ним на первом этапе инженер по знаниям создает онтологию предметной области (или онтологию знаний), используя язык описания онтологии, на втором – автоматически генерируется редактор знаний (по онтологии), и затем эксперт предметной области (возможно, с инженером по знаниям) формирует базу знаний [14–17]. При этом метод генерации интерфейса редактора по созданной онтологии является фиксированным. Таким образом, основной недостаток описанного подхода состоит в том, что интерфейс генерируется на основе онтологии по «жесткой схеме», встроенной в генератор редакторов (рис. 1).

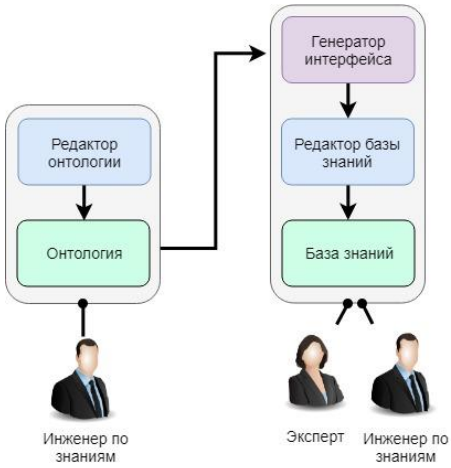

Рис. 1. Традиционный подход к генерации редакторов знаний
Fig. 1. The traditional approach to generating knowledge editors

Для устранения указанного недостатка предлагается новая концепция, основные положения которой заключаются в следующем:

1. Вводится множество абстрактных элементов интерфейса, т.е. элементов, которые решают определенные интерфейсные задачи (ввод строки или числа, выбор из множества, вывод строки или числа и т.п.), но не фиксируют конкретный интерфейсный элемент и его визуальное представление.

2. Для каждого абстрактного элемента интерфейса задается множество его адаптаций – возможных вариантов визуального представления. Каждому визуальному представлению соответствует некоторый элемент WIMP-интерфейса с набором параметров. При этом с каждым элементом адапта-

[1] От англ. usability – удобство использования, пригодность использования, эргономичность – способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях (ГОСТ Р ИСО/МЭК 25010–2015).

ции (WIMP-элементом) связано условие его применимости, которое зависит от типа операционной системы, устройства, структуры информации, типа пользователя, его предпочтений, среды использования и других факторов. Учитывая, что интерфейсные элементы, знания о проектировании, платформы и устройства постоянно изменяются и должны непрерывно совершенствоваться, соответствие между абстрактными интерфейсными элементами и их адаптациями задается в базе знаний о проектировании интерфейса.

3. Модель интерфейса (модель задач). Модель интерфейса определяет иерархию, расположение и порядок отображения (абстрактных) интерфейсных элементов в процессе работы с сервисом с учетом выполнения программной логики. Модель интерфейса определяет предпочтения пользователя (интерфейсный стиль, тип диалога, уровень владения компьютером и пр.). Вся информация, полученная либо от пользователя, либо в процессе его работы с редактором (число ошибок, скорость работы, порядок внесения информации и др.) сохраняется. Если пользователь не задает начальную информацию, то используется система умолчаний, а адаптация осуществляется на основе динамической (полученной в процессе работы с пользователем) информации.

4. Разрабатывается генератор интерфейса, который по онтологии предметной области, базе знаний о проектировании интерфейса и модели интерфейса формирует пользовательский интерфейс и обеспечивает его адаптацию к пользователю в процессе формирования базы знаний.

В соответствии с приведенными выше положениями концепции и современным подходом к разработке СБЗ необходимы следующие информационные ресурсы:

1. Онтология WIMP-элементов и сформированная по ней база (библиотека) WIMP-элементов, содержащая повторно используемое и расширяемое множество WIMP-элементов и их атрибутов, зависящих от платформы и устройства, а также специального (пользовательского) множества WIMP-элементов – реализаций для конкретной предметной области.

2. Онтология знаний о проектировании интерфейсов и база знаний о проектировании интерфейсов, соответствующая стандартам проектирования.

3. Онтология модели интерфейса.

2. Архитектура программного средства

В соответствии с приведенной концепцией предлагается следующая схема генерации WIMP-интерфейсов редакторов знаний (рис. 2).

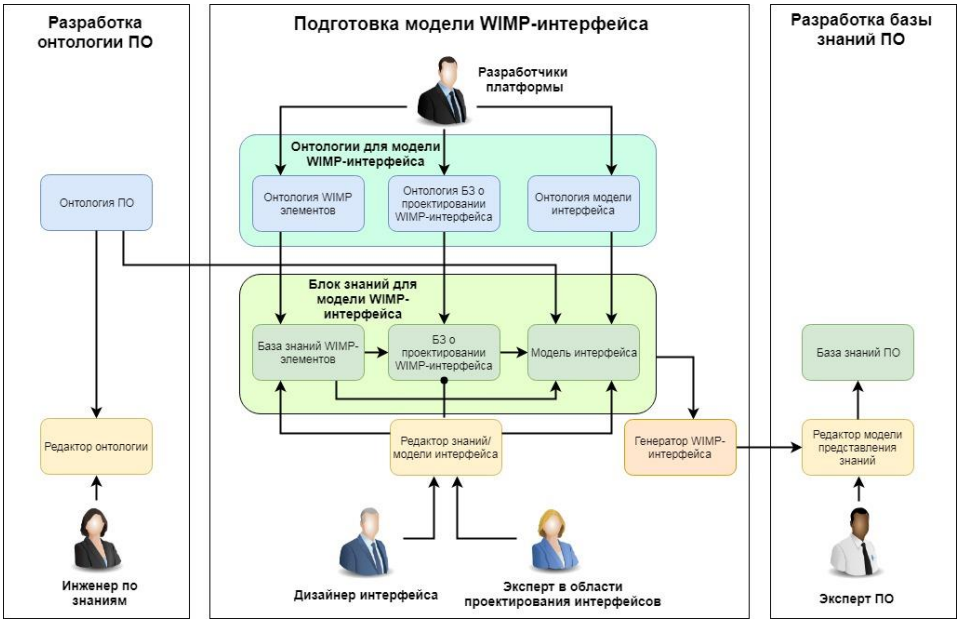

Рис. 2. Формирование редактора с адаптивным WIMP-интерфейсом
Fig. 2. Formation of an editor with adaptive WIMP-interface

Формирование онтологии производится, как и в традиционном способе. Далее эксперт предметной области формирует модель интерфейса (указывает предпочтения пользователя, уровень владения компьютером, предпочтительный тип диалога и др.) либо дизайнер интерфейса задает в модели интерфейса общепринятые в предметной области способы представления информации. Этот этап может отсутствовать, тогда используется система умолчаний, при этом в процессе работы пользователя с редактором знаний модель интерфейса дополняется динамической информацией о пользователе (скорость работы, число ошибок, порядок действий и др.). Генератор редактора знаний по онтологии, модели интерфейса и базе знаний о проектировании интерфейса формирует пользовательский интерфейс и управляет им в процессе формирования базы знаний. Отдельным блоком системы является формирование базы знаний о проектировании интерфейса. База знаний разрабатывается на основе онтологии базы знаний по технологии создания онтологий, при этом используется библиотека WIMP-элементов, которая также формируется по соответствующей онтологии.

3. Онтология WIMP-элементов

Для формального представления онтологий будет использоваться нотация языка ИРУО [17–18].

Онтология WIMP-элементов состоит из двух основных разделов: множество WIMP-элементов и множество стилей:

Онтология WIMP элементов {
~copy WIMP-элементы {*~ set WIMP -элемент* {
~copy CSS-стили
<Набор уникальных свойств>
}}
~copy CSS-стили {*~set CSS* {
~copy selector {*<...>*}
~copy body {*<...>*}
}}

WIMP-элементы – это стандартные «оконные» интерфейсные элементы: Базовый элемент (Div), Текст (Text), Кнопка (Button), Текстовое поле ввода, Списки (пролистываемый, выпадающий) (List, Combobox), Вкладки (Tabs), Радио-кнопки (RadioGroup), Чек-боксы (CheckboxGroup), Ползунок (Slider), Пиктограмма (Picture). Структурно каждый из этих элементов состоит из некоторого набора уникальных для данного элемента атрибутов и стилей CSS.

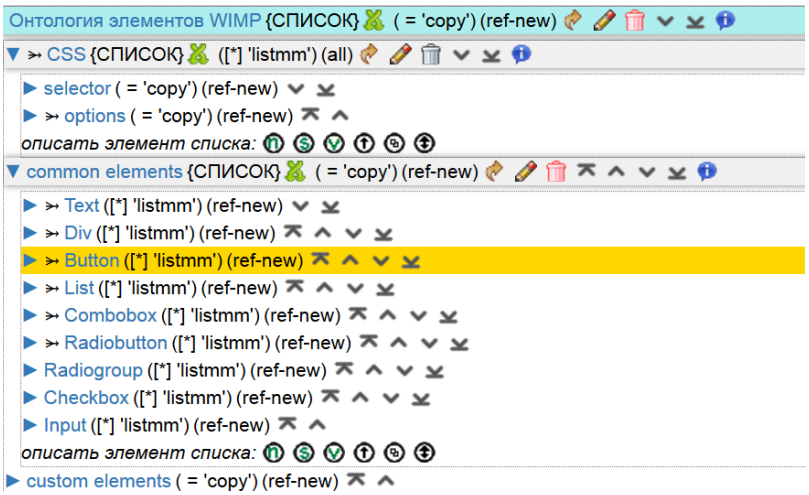

Рис. 3. Скриншот онтологии WIMP-элементов
Fig. 3. Screenshot of the ontology of WIMP-elements

CSS-стили – это каскадные таблицы стилей (аналогичные таблицам Cascading Style Sheets – CSS из веб-технологий), которые задают множества возможных типов оформления внешнего вида

WIMP-элементов, начиная от простых, таких как шрифт, цвет, размер элементов, до составных, содержащих наборы интерфейсных решений. Их определение осуществляется в соответствии принятым определением в веб-разработке [19] и состоит из «селектора» и «тела». «Селектор» определяет, для каких именно элементов интерфейса применить заданные правила (свойства). «Тело» стиля содержит набор свойств, которые будут применяться для соответствующих селектору элементов (рис. 3).

4. Онтология БЗ о проектировании WIMP-интерфейса

База знаний о проектировании WIMP-интерфейса строится на основе онтологии, которая задает наборы соответствий между абстрактным интерфейсным элементом и его возможными представлениями в интерфейсе (рис. 4).

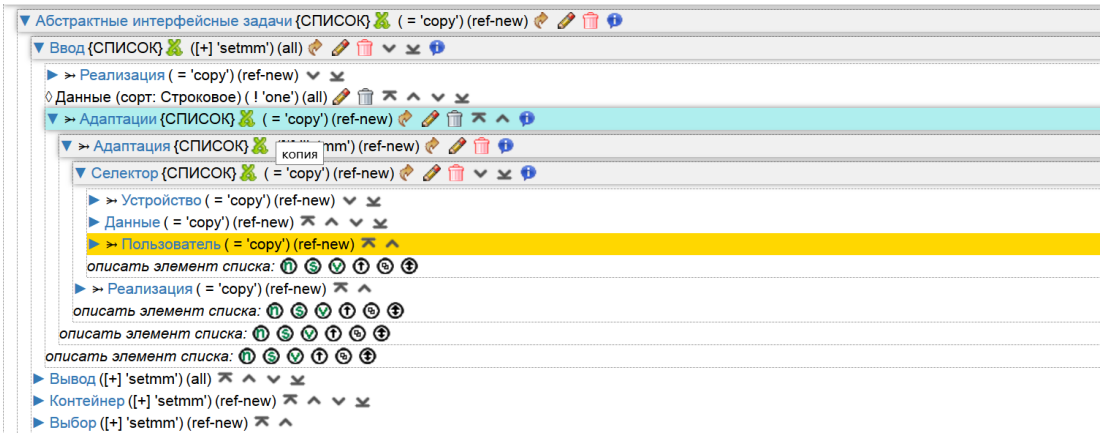

Рис.4. Скриншот онтологии с базами знаний о проектировании WIMP-интерфейса
Fig. 4. Screenshot of an ontology with knowledge bases on designing a WIMP-interface

Применимость элемента определяется набором условий (адаптаций).

~one Абстрактный интерфейсный элемент {

~one -> WIMP-элемент

~set <Адаптация>

}

WIMP-элемент – это ссылка на WIMP-элемент (с конкретными параметрами) из онтологии WIMP.

<Адаптация> – это множество вариантов применимости интерфейсного элемента в зависимости от условий:

~set Адаптация {

~copy <Селектор адаптации>

~copy < Реализация>

}

<Селектор адаптации> – это набор условий адаптации, при выполнении которых интерфейсный элемент получает параметры, указанные в блоке <Реализация>. Условия реализации определяются следующими факторами: устройство, данные, пользователь:

~copy Селектор адаптации {

~copy <Устройство>

~copy <Данные>

~copy <Пользователь>

}

<Устройство> определяет его тип (например, ПК, планшет, смартфон), операционную систему (например, Android, Windows, IOS) и размеры.

<Данные> содержит параметры, характеризующие внешнее или внутреннее окружение элемента, которое может влиять на представление данного элемента, например количество потомков.

<Пользователь> определяет параметры адаптации, зависящие от данных пользователя, который работает в настоящее время с интерфейсом. Данные пользователя могут быть динамическими (скорость работы, количество ошибок и др.) и статическими (возраст, особые запросы и особенности и др.).

С каждым абстрактным элементом интерфейса может быть связано множество возможных интерфейсных элементов, как определенных разработчиками системы, так и разработанных пользователями. Список таких элементов был сформирован авторами данной работы на основе анализа литературы [2–4] и собственного опыта проектирования интеллектуальных интерфейсов [16–18]. Каждый абстрактный элемент задается своим именем и набором параметров (рис. 5):

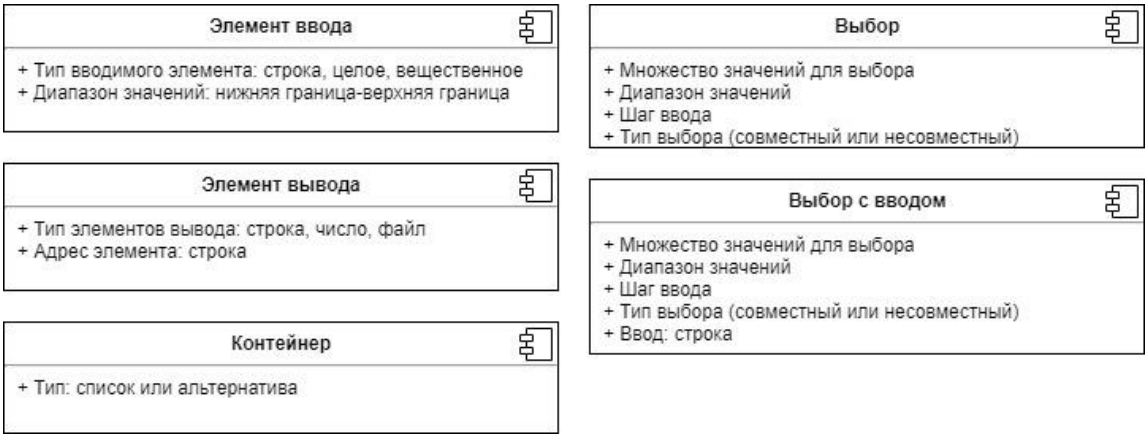

Рис. 5. Абстрактные элементы интерфейса
Fig. 5. Abstract interface elements

На рис. 6 приведен пример возможных реализаций для абстрактного элемента «ВЫБОР». Как видно из данного примера, выбор элементов может осуществляться с использованием нескольких возможных элементов WIMP-интерфейса: группой радиокнопок, группой чек-боксов, пролистываемым и раскрывающимся списками. Выбор конкретного элемента WIMP-интерфейса определяется данными (количеством элементов и особенностью их выбора – несовместный (единственный) выбор или совместный), типом пользователя (продвинутый или новичок) и т.д.

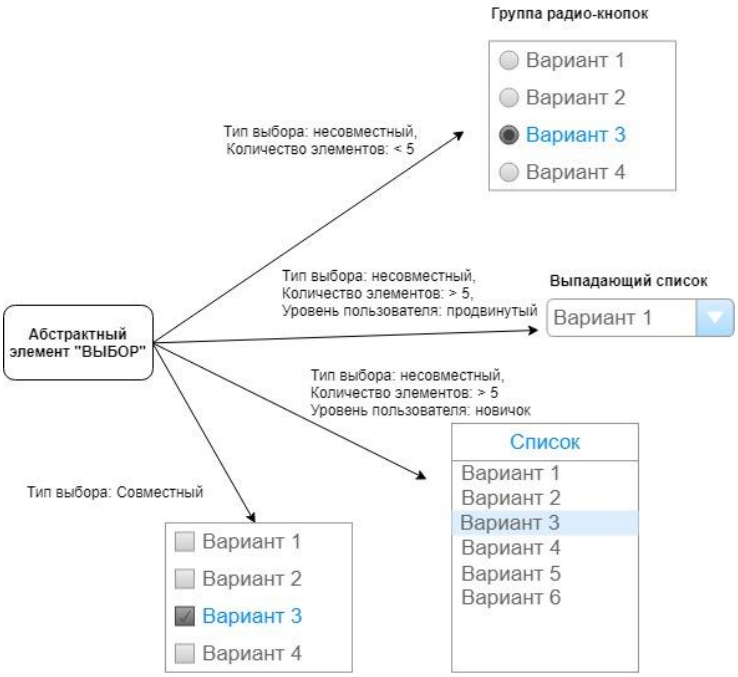

Рис. 6. Возможные реализации WIMP-элементов для абстрактного интерфейсного элемента «ВЫБОР»
Fig. 6. Possible implementations of WIMP-elements for the "SELECT" abstract interface element

Абстрактным элементам соответствует не только множество конкретных WIMP-элементов, но и множество вариантов реализации каждого интерфейсного элемента для различных операционных систем и устройств (размеры экрана, возраст пользователя и др.) – адаптации. Если ни одна адаптация не удовлетворяет условиям, то необходим отдельный вариант адаптации интерфейсного элемента – некоторый WIMP-элемент по умолчанию. Интерфейсный элемент по умолчанию – это ссылка на любой описанный WIMP-элемент (с конкретными параметрами) из библиотеки WIMP-элементов.

Условия в селекторе адаптации имеют свой приоритет. Приоритеты могут иметь значение в случае, когда выполнились все условия для двух или более селекторов, содержащих некоторые одинаковые параметры изменения в реализации WIMP-элемента. В таком случае будет выбран тот параметр, селектор которого имеет больший суммарный приоритет: приоритет селектора равен сумме приоритетов всех его условий.

5. Онтология модели интерфейса

Онтология модели интерфейса хранит информацию следующих видов:

1. Пользовательская (информация, полученная от пользователя).
2. Динамическая (информация, полученная в процессе работы с сервисом).

```
Онтология базы знаний о проектировании WIMP-интерфейса {
    ~copy Пользователь {    ... }
    ~copy Динамическая информация { ... }
}
```

Модель интерфейса определяет иерархию, расположение и порядок отображения (абстрактных) интерфейсных элементов в процессе работы с сервисом с учетом выполнения программной логики. Определяется отображение и поведение интерфейса в соответствии с индивидуальными характеристиками и предпочтениями пользователя сервиса.

Пользователь сам заполняет данные о себе, вносит предпочтения. Если пользователь не задает начальную информацию, то используются параметры по умолчанию. Модель пользователя включает следующие компоненты: пол, возраст, образование и др.

```
~copy Пользователь {
        ~one -> Пол {Муж | Жен}
        ~one -> Возраст [integer]
        ~set -> Параметр {
                ~one -> Ключ [string]
                ~one -> Значение [string]
    }
  ...
}
```

Динамическая информация включает следующие виды данных: число ошибок, скорость работы, порядок внесения информации и др.

```
~copy Динамическая информация {
        ~one -> Число ошибок [integer]
        ...
}
```

Заключение

Предложена концепция создания адаптивных генераторов пользовательских WIMP-интерфейсов по онтологии знаний. В статье описаны необходимые в концепции онтологии (онтология WIMP-элементов, онтология знаний о проектировании интерфейсов и онтология модели интерфейса) и соответствующие им базы знаний. Основной результат работы заключается в модификации

подхода к формированию редакторов баз знаний и разработке концепции автоматизированного создания WIMP-интерфейса с использованием введенных абстрактных адаптивных элементов интерфейса, соотносящихся с элементами метаязыка и отображаемыми WIMP-элементами. В результате вместо фиксированного метода генерации интерфейса редактора предлагается адаптивный механизм, автоматически учитывающий изменяющиеся параметры реализации интерфейса: устройство, структуру информации, предпочтения пользователя.

ЛИТЕРАТУРА

1. Myers B.A., Rosson M.B. Survey on user interface programming // Proc. SIGCHI'92: Human Factors in Computing Systems. Monterrey, CA, May 3–7, 1992. P. 195–202.
2. Корончик Д.Н. Пользовательские интерфейсы интеллектуальных систем // Кибернетика и программирование. 2012. № 1. С. 16–22.
3. Белоусова С.А., Рогозов Ю.И. Анализ подходов к созданию пользовательского интерфейса // Известия Южного федерального университета. Технические науки. 2014. № 6 (155). С. 142–148.
4. Верлань А.Ф., Сопель М.Ф., Фуртат Ю.О. Об организации адаптивного пользовательского интерфейса в автоматизированных системах // Известия Южного федерального университета. Технические науки. 2014. № 1 (150). С. 100–110.
5. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб. : Питер, 2001. 384 с.
6. Грибова В.В., Клещев А.С. Управление проектированием и реализацией пользовательского интерфейса на основе онтологий // Проблемы управления. 2006. № 2. С. 58–62.
7. Загорулько Ю.А., Загорулько Г.Б. Онтологии и их практическое применение в системах, основанных на знаниях // Всероссийская конференция с международным участием «Знания – Онтологии – Теории» (ЗОНТ-2011). Новосибирск, 3–5 окт. 2011. Новосибирск : Ин-т математики им. С.Л. Соболева СО РАН, 2011. С. 132–141
8. Ломов П.А., Шишаев М.Г., Диковицкий В.В. Преобразование OWL-онтологий для визуализации и использования в качестве основы пользовательского интерфейса // Онтология проектирования. 2012. № 3 (5). С. 49–61.
9. Подлипский О.К. Построение баз знаний группой экспертов // Компьютерные исследования и моделирование 2010 Т. 2, № 1. С. 3–11.
10. Щеглов С.Н. Онтологический подход и его использование в системах представления знаний // Известия Южного федерального университета. 2009. № 4 (93) С. 146–153.
11. Bechhofer S., Stevens R., Ng G., Jacoby A., Goble C. Guiding the user: an ontology driven interface // Proceedings of the User Interfaces to Data Intensive Systems (UIDIS). 1999. P. 158–161.
12. Feikje H., Chris M., Peter E. Evaluating an ontology-driven WYSIWYG interface // Proc. of the 5th Int. Conf. on Natural Language Generation. 2008. P. 138–146.
13. Musen M.A. Dimensions of knowledge sharing and reuse // Computers and Biomedical Research. 1992. V. 25. P. 435–467.
14. Gribova V., Kleschev A., Moskalenko P., Timchenko V., Fedorischev L., Shalfeeva E. The IACPaaS cloud platform: Features and perspectives // Second Russia and Pacific Conference on Computer Technology and Applications (RPC) (Vladivostok, Russia, 25–29 Sept. 2017). IEEE. 2017. P. 80–84.
15. Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Ч. 1. Разработка базы знаний и решателя задач // Программная инженерия. 2015. № 12. С. 3–11.
16. Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А. Модель редактора сложноструктурированных информационных единиц, управляемого метаинформацией, и его реализация // Научно-техническая информация. Сер. 2. 2016. № 2. С. 1–13.
17. Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А. Модель порождения орграфов информации по орграфу метаинформации для двухуровневой модели сложноструктурированных информационных единиц // Научно-техническая информация. Сер. 2. 2015. № 12. С. 26–38.
18. Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А. Двухуровневая модель сложноструктурированных информационных единиц, соответствующая метафоре анкетирования // Научно-техническая информация. Сер. 2. 2015. № 10. С. 1–10.
19. CSS. URL: https://ru.wikipedia.org/wiki/CSS (дата обращения: 01.02.2019).

Поступила в редакцию 25 февраля 2019 г.

Gribova V.V., Fedorischev L.A. (2019) ADAPTIVE GENERATOR WIMP-INTERFACE FOR KNOWLEDGE BASE EDITOR BASED ON AN ONTOLLOGY. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]*Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 49. pp. 110–119

DOI: 10.17223/19988605/49/13

The concept of creating adaptive generators of user-defined WIMP-interfaces on knowledge ontology is proposed. In accordance with this concept, a domain ontology is created, a knowledge editor is automatically generated, and then a domain expert creates a knowledge base. The article presents the ontologies necessary in the concept (the ontology of WIMP elements, the ontology of knowledge about interface design and the ontology of the interface model) and the corresponding knowledge bases.

The Ontology of WIMP elements consists of two main sections: the set of WIMP elements and the set of CSS styles. WIMP elements are standard "window" interface elements: buttons, text fields, controls, containers, and others. CSS styles are cascading style sheets that define a set of possible types of appearance design for WIMP elements.

The knowledge base on designing a WIMP interface is built on the basis of an ontology that defines sets of correspondences between an abstract interface element and its possible representations in the interface. The applicability of an element is determined by a set of conditions (adaptations).

The ontology of the interface model stores information of both the user and dynamic information obtained in the process of working with the service.

The novelty of the research lies in the modification of the approach to the formation of knowledge base editors and the development of the concept of the automated creation of a WIMP interface using the newly introduced abstract adaptive interface elements that correlate with the elements of the metalanguage and the displayed WIMP elements. As a result, instead of the fixed method of generating the interface of the editor, a new adaptive mechanism is proposed that automatically takes into account the changing parameters of the interface implementation: device, information structure, user preferences. To illustrate the proposed ideas, conceptual schemes for generating the interface are given and relevant examples are presented.

Keywords: WIMP-interface; ontologies; adaptability; knowledge base editor.

GRIBOVA Valerya Victorovna (Doctor of Technical Sciences, Institute of Automation and Control Processes, Far Eastern Branch of Russian Academy Sciences, Vladivostok, Russian Federation).
E-mail: gribova@iacp.dvo.ru

FEDORISCHEV Leonid Aleksandrovich (Candidate of Technical Sciences, Institute of Automation and Control Processes, Far Eastern Branch of Russian Academy Sciences, Vladivostok, Russian Federation).
E-mail: fleo1987@mail.ru

REFERENCES

1. Myers, B.A. & Rosson, M.B. (1992) Survey on user interface programming. *Proc. SIGCHI'92: Human Factors in Computing Systems.* Monterrey. CA. May 3–7. pp. 195–202.
2. Koronchik, D.N. (2012) Intelligent system user interfaces. *Kibernetika i programmirovaniye*. 1. pp. 16–22. DOI: 10.7256/2306-4196.2012.1.13861
3. Belousova, S.A. & Rogozov, Yu.I. (2014) Analysis of approaches to user interface building. *Izvestiya Yuzhnogo federalnogo universiteta. Tekhnicheskiye nauki – Izvestiya SFedU. Engineering Sciences*. 6(155). pp. 142–148.
4. Verlan, A.F., Sopel, M.F. & Furtat, Yu.O. (2014) On adaptive user interface organization in automated systems. *Izvestiya Yuzhnogo federalnogo universiteta. Tekhnicheskiye nauki – Izvestiya SFedU. Engineering Sciences.* 1(150). pp. 100–110.
5. Gavrilova, T.A. & Khoroshevsky, V.F. (2001) *Bazy znaniy intellektual'nykh sistem* [Intelligent systems knowledge bases]. St. Petersburg: Piter.
6. Gribova, V.V. & Kleshchev, A.C. (2006) Upravlenie proektirovaniem i realizatsiey pol'zovatel'skogo interfeysa na osnove ontologiy [Control of ontology-based user interface design and implementation]. *Problemy upravleniya – Control Sciences*. 2. pp. 58–62.
7. Zagorulko, Yu.A. & Zagorulko, G.B. (2011) [Ontologies and their practical application in knowledge-based systems]. *Znaniya – Ontologii – Teorii* [Knowledge-Ontologies-Theories]. Proc. of the All-Russian Conference with International Participation. Novosibirsk.
8. Lomov, P.A., Shishaev, M.G. & Dikovitsky, V.V. (2012) Transformation of owl-ontology for visualization and usage as the basis of the user interface. *Ontologiya proektirovaniya – Ontology of Designing.* 3(5). pp. 49–61.
9. Podlipsky, O.K. (2010) Construction of knowledge bases by a group of experts. *Kompyuternye issledovaniya i modelirovanie – Computer Research and Modeling.* 2(1). pp. 3–11. DOI: 10.20537/2076-7633-2010-2-1-3-11
10. Shcheglov, S.N. (2009) The ontologic approach and its use in systems of representation of knowledge. *Izvestiya Yuzhnogo federalnogo universiteta*. 4(93). pp. 146–153.
11. Bechhofer, S., Stevens, R.D., Ng, G., Jacoby, A. & Goble, C. (1999) Guiding the user: An ontology driven interface. *UIDIS.* pp. 158–161. DOI: 10.1109/UIDIS.1999.791472
12. Feikje, H., Chris, M. & Peter, E. (2008) Evaluating an ontology-driven WYSIWYG interface. *Proc. of the 5th Int. Conf. on Natural Language Generation*. pp. 138–146.
13. Musen, M.A. (1992) Dimensions of knowledge sharing and reuse. *Computers and Biomedical Research.* 25. pp. 435–467. DOI: 10.1016/0010-4809(92)90003-S

14. Gribova, V., Kleschev, A., Moskalenko, P., Timchenko, V., Fedorischev, L. & Shalfeeva, E. (2017) The IACPaaS Cloud Platform: Features and Perspectives. *Second Russia and Pacific Conference on Computer Technology and Applications (RPC).* Vladivostok. Russia. September 25–29, 2017. IEEE. pp. 80–84. DOI: 10.1109/RPC.2017.8168076
15. Gribova, V.V., Kleshchev, A.S., Krylov, D.A., Moskalenko, F.M., Timchenko, V.A. & Shalfeeva, E.A. (2015) A Base Technology for Development of Intelligent Services with the Use of IACPaaS Cloud Platform. Part 1. A Development of Knowledge Base and Problem Solver. *Programmnaya inzheneriya – Software Engineering*. 12. pp. 3–11.
16. Gribova, V.V., Kleshchev, A.S., Moskalenko, F.M. & Timchenko, V.A. (2016) Model' redaktora slozhnostrukturirovannykh informa-tsionnykh edinits, upravlyaemogo metainformatsiey, i ego realizatsiya [Editor's model for complex structured information units driven by meta-information and its implementation]. *Nauchno-tekhnicheskaya informatsiya*. *Series 2*. 2. pp. 1–13.
17. Gribova, V.V., Kleshchev, A.S., Moskalenko, F.M. & Timchenko V.A. (2015) Model' porozhdeniya orgrafov informatsii po orgrafu metainformatsii dlya dvukhurovnevoy modeli slozhnostrukturirovannykh informatsionnykh edinits [The model of generating digraphs of information on the digraph of meta-information for a two-level model of complex structured information units]. *Nauchno-tekhnicheskaya informatsiya*. *Ser. 2*. 12. pp. 26–38.
18. Gribova, V.V., Kleshchev, A.S., Moskalenko, F.M. & Timchenko, V.A. (2015) Dvukhurovnevaya model' slozhnostrukturirovannykh informatsionnykh edinits, sootvetstvuyushchaya metafore anketirovaniya [A two-level model of complex structured information units corresponding to the questionnaire metaphor]. *Nauchno-tekhnicheskaya informatsiya. Ser. 2*. 10. pp. 1–10.
19. CSS. (n.d.) [Online] Available form: https://ru.wikipedia.org/wiki/CSS (Accessed: 1st February 2019).