

УДК 004.514.6

## Онтологический подход к генерации адаптивных WIMP-интерфейсов редакторов баз знаний<sup>1</sup>

В.В. Грибова (*gribova@dvo.ru*),

Л.А. Федорищев (*fleo1987@mail.ru*)

Институт автоматики и процессов управления ДВО  
РАН, Владивосток

В работе представлен подход к генерации адаптивных WIMP-интерфейсов редакторов знаний по онтологии предметной области и базе знаний о правилах их проектирования. Рассмотрены онтологии WIMP-элементов, базы знаний о проектировании интерфейсов, модели интерфейса. Представлен краткий алгоритм генерации интерфейса базы знаний.

**Ключевые слова:** пользовательский интерфейс, онтологии, базы знаний, адаптивность, WIMP-интерфейс, автоматическая генерация

### Введение

Редакторы баз знаний являются одним из классов программных систем, поэтому "дружественный" пользовательский интерфейс, соответствующий требованиям пользователей, особенностям представления информации в предметной области, среде использования является одним из ключевых факторов, по которым пользователи принимают решение об использовании того или иного редактора [Сегун 2012]. Несмотря на то, что редактор баз знаний - это один из классов программных систем, существующие подходы в области автоматизации адаптивных интерфейсов для них не подходят. Редакторы баз знаний характеризуются следующими особенностями, отличающими их от интерфейсов программных систем других типов: большой размер баз знаний, часто состоящий из тысяч и даже десятков тысяч терминов, их характеристик и значений; большая связанность информационных ресурсов (данных и сложно-структурированных знаний) между собой; универсальность - редакторы баз знаний относятся к универсальным, проблемно и задаче независимым программным продуктам, соответственно, в них должны быть заложены общие принципы формирования интерфейсов, в то время

---

<sup>1</sup> Работа выполнена при частичной финансовой поддержке РФФИ (проект № 17-07-00956 и 16-07-00340) и «ДВ» 18-5-078, 18-5-060

как для каждой программной системы специально разрабатывается пользовательский интерфейс, ориентированный на конкретную функциональность, в соответствии с ней выделяются задачи пользователей, строится ориентированный на функциональность сценарий диалога.

Современный подход к построению баз знаний основан на использовании онтологий [Щеглов 2009; Гаврилова и др, 2016]. Редакторы баз знаний, основанные на данном подходе (Protege, IWE, OntoEdit, GrOWL, GraphI, RDFGravity, WebVOWL, Ontolingua, OntoSaurus, OilEd, WebOnto, WebODE, SmartTools, Semp, CLIPS и др.), имеют средства генерации интерфейсов для редактирования знаний по онтологии на основе фиксированной модели генерации интерфейсов. Понимая важность интерфейсной проблемы, в некоторых редакторах реализуется несколько моделей визуализации знаний, что позволяет просмотреть их в различных формах.

Вместе с тем эксперты, формирующие базы знаний, имеют разный уровень владения компьютером, в предметных областях устоялись и используются свои стандарты к представлению информации, сценарию человеко-компьютерного взаимодействия, что требует гибких подходов к построению пользовательского интерфейса. В связи с вышесказанным, разработка моделей и методов генерации адаптивных WIMP<sup>2</sup> - интерфейсов (наиболее востребованных пользователями именно для ввода и редактирования знаний), является актуальной задачей [Macik 2016].

В данной работе предлагается подход к автоматической генерации адаптивных WIMP-интерфейсов редакторов баз знаний по онтологии предметной области и базе знаний о правилах их проектирования.

## **1. Подход к генерации адаптивных WIMP-интерфейсов**

Область знаний, связанная с разработкой интерфейсов, постоянно изменяется - появляются новые интерфейсные элементы и платформы, расширяются свойства существующих элементов, совершенствуются требования к их проектированию [Macik 2014, Белоусова и др., 2014]. В связи с этим важно, чтобы предложенные методы были развиваемыми. В качестве реализации этого основного требования предлагается использовать онтологический подход для формирования знаний о проектировании интерфейсов.

Основная идея предложенного подхода заключается в том, что пользовательский интерфейс автоматически генерируется по модели знаний (онтологии), базе знаний о проектировании интерфейсов, а также модели интерфейса, в которой определены специфические для данного

---

<sup>2</sup> WIMP - Windows, Icons, Menu, Pointing Device

интерфейса параметры генерации (она может отсутствовать, в этом случае используется система умолчаний).

Основными элементами предложенного подхода являются:

- *множество абстрактных интерфейсных элементов* (например, выбор элемента из множества, вывод элемента и др.), задающих возможные действия пользователя либо команды интерфейса; абстрактные элементы не зависят от платформы и конкретных WIMP-элементов; каждому абстрактному элементу интерфейса ставится в соответствие конструкция языка для описания модели знаний (онтологии);

- *онтология WIMP-элементов*, в соответствии с которой формируется библиотека WIMP-элементов;

- *онтология базы знаний о проектировании интерфейса*, и, *база знаний, сформированная на ее основе*. Каждому элементу абстрактного интерфейса в базе знаний сопоставляется множество конкретных элементов интерфейса из библиотеки WIMP-элементов, определенных реализацией, и множество условий, которые задают правила применимости этих элементов. Условия определяются требованиями юзабилити<sup>3</sup>, характеристиками платформы и профиля пользователя.

- *онтология модели интерфейса*, описывающая структуру параметров, необходимую для генерации адаптивного интерфейса конкретного редактора базы знаний. По онтологии формируется *модель интерфейса*. Параметры могут быть статическими (уровень владения компьютером, платформа, возраст) и динамическими, которые определяются в процессе взаимодействия пользователя с интерфейсом (число ошибок, время ввода/выбора элементов и др.). Помимо этого в модели интерфейса возможно задать конкретные представления в интерфейсе для выбранных элементов онтологии базы знаний.

- *генератор интерфейса*, который по модели знаний (онтологии), базе знаний о проектировании интерфейса и модели интерфейса формирует пользовательский интерфейс редактора базы знаний.

Предложенный подход реализуется на облачной платформе IACPaas, предназначенной для создания интеллектуальных систем [Gribova et al., 2017]. Базы знаний создаются на основе двухуровневого подхода: на языке ИРУО формируется метаинформация (онтология), задающая структуру базы знаний и ее терминологию, а также правила формирования и ограничения, а затем по ней формируется база знаний [Gribova 2015]. Каждому элементу языка ИРУО поставлен в соответствие элемент абстрактного интерфейса. Принципиально важно, что при таком подходе в модель знаний (онтологию) не включается информация об интерфейсе,

---

<sup>3</sup> от англ. usability — «удобство и простота использования, степень удобства использования»

что соответствует современному требованию к разработке программных систем.

На рис. 1 представлена схема реализации предложенного подхода.

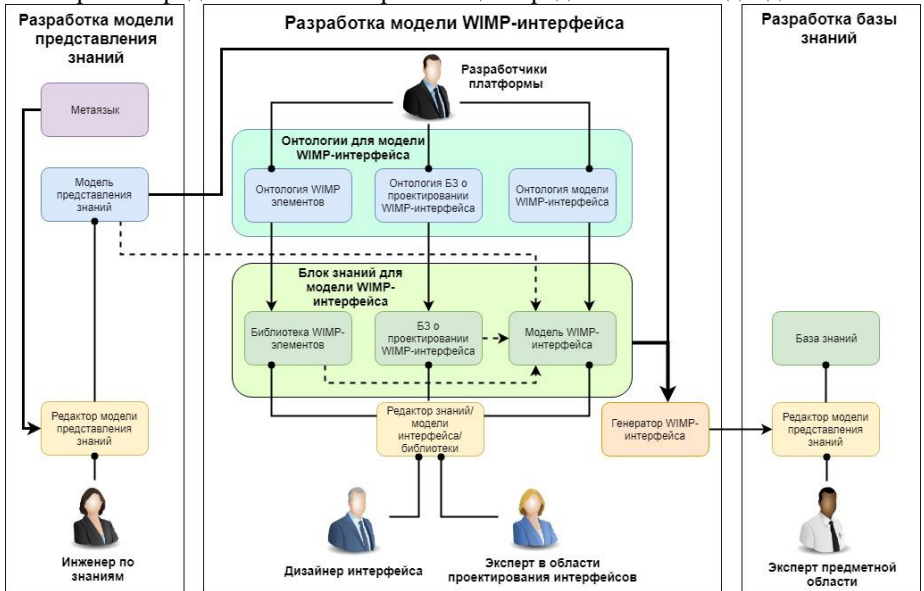


Рис. 1. Схема реализации подхода к автоматической генерации интерфейса

## 2. Онтология WIMP-элементов

Библиотека WIMP-элементов формируется на основе онтологии WIMP-элементов. Она включает описание структуры стилей и описание структуры WIMP-элементов.

```

Онтология WIMP элементов {
  ~сору CSS-стили {
    <...>
  }
  ~сору WIMP-элементы {
    <...>
  }
}

```

Группа *<~сору CSS-стили>* - это множество стилей, подобных традиционным стилям CSS, использующимся в веб-разработке. Стили определяют оформление внешнего вида интерфейсных элементов,

начиная от простых, таких как шрифт, цвет, размер элементов, до составных, содержащих наборы интерфейсных .

Группа *<~copy WIMP-элементы>* - это группа стандартных элементов интерфейса, таких как кнопка, поле ввода, список и др..

```
~copy WIMP-элементы {  
  ~ set Text {<...>}  
  ~ set Button {<...>}  
  ...  
}
```

Каждый элемент интерфейса состоит из набора атрибутов и стилей CSS. Например:

*<set Text>* - это множество текстовых WIMP-элементов с параметром *<text>* и стилями CSS:

```
~ set Text: {  
  ~set -> CSS  
  ~one text  
}
```

*<set Button >* - это множество кнопок. Каждая кнопка имеет надпись и состояния: default, clicked, focused, disabled, mouseover, mousedown.

```
~ set Button: {  
  ~set -> CSS  
  ~one text  
  ~copy default { ~set -> CSS }  
  <...>  
}
```

Данная онтология в настоящее время реализована на облачной платформе IACPaaS.

### 3. Онтология знаний 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. 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 {
    ~сору <Селектор адаптации>
    ~сору <Реализация>
}

```

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

```

~сору Селектор адаптации {
    ~сору <Устройство>
    ~сору <Среда>
    ~сору <Пользователь>
}

```

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

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

<Пользователь> определяет параметры адаптации, зависящие от профиля пользователя, например, возраст, уровень владения компьютером и др.

#### 4. Модель WIMP-интерфейса

База знаний о проектировании WIMP-интерфейса позволяет генерировать адаптивные интерфейсы, зависящие от предметной области, пользователя и других параметров. В модели интерфейса задаются конкретные значения параметров, которые влияют на генерацию интерфейса. Некоторые параметры определяются автоматически, например, платформа и устройство, а возраст, уровень владения компьютером, влияющие на выбор конкретного элемента, их размер, шрифт задает пользователь (иначе выбираются значению "по умолчанию"). Некоторые параметры, которые влияют на представление, например, число ошибок, скорость работы, уточняются в процессе взаимодействия с пользователем и заносятся в модель интерфейса автоматически для контекстной адаптации интерфейса. Также пользователь может сам выбрать конкретное представление в интерфейсе для тех или иных элементов онтологии знаний, для которой генерируется интерфейс:

```

~set Элемент {
    ~one -> Язык ИРУО
    ~one -> WIMP-элемент
}

```

## 5. Генерация интерфейса

Алгоритм генерации заключается в следующем:

- 1) Получить на вход онтологию модели знаний, каждому элементу онтологии сопоставить конструкцию языка ИРУО;
- 2) По таблице соответствий найти абстрактный интерфейсный элемент, который соответствует этой конструкции языка;
- 3) По базе знаний выбрать множество адаптаций для абстрактного элемента;
- 4) Среди множества адаптаций выбрать ту, которая соответствует модели интерфейса;
- 5) Отобразить в интерфейсе выбранную адаптацию.

На рис. 2 приводится пример генерации WIMP-интерфейса редактора базы знаний по лечению заболеваний с использованием предложенной технологии.

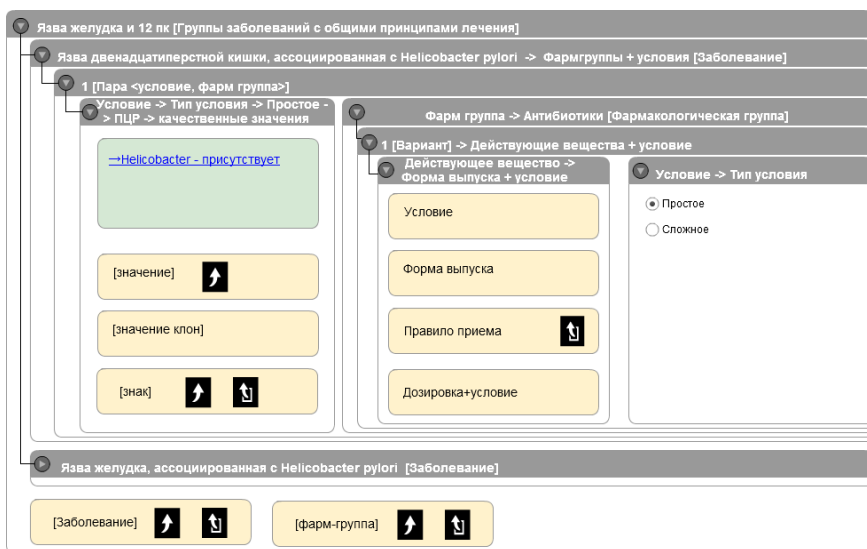


Рис. 2 Пример сгенерированного интерфейса для редактора знаний

## Заключение

В работе описан новый подход к генерации адаптивных WIMP-интерфейсов редакторов баз знаний. Генерация осуществляется по модели знаний (онтологии), модели интерфейса и базе знаний о проектировании интерфейса. Прототипная версия предложенного подхода реализована на облачной платформе IASaaS, редактор знаний также является облачным

(доступен через Интернет). Возможно использование данного подхода и на других программных платформах. Для этого необходимо определить соответствие между конструкциями языка представления знаний и абстрактными элементами интерфейса. Важной особенностью предложенного подхода является также то, что знания о проектировании интерфейса выделены в отдельный компонент, по мере появления новых интерфейсных элементов, знаний о проектировании интерфейсов, база знаний может быть расширена.

### Список литературы

- [Cerny et al., 2012] T. Cerny, V. Chalupa, and M. J. Donahoo. Towards smart user interface design. In Info. Science and Applications (ICISA), 2012 Int. Conf. on, pages 1–6. IEEE
- [Щеглов, 2009] Щеглов С.Н. Онтологический подход и его использование в системах представления знаний// Известия Южного федерального университета. – 2009
- [Гаврилова и др, 2016] Гаврилова Т.А., Кудрявцев Д.В., Муромцев Инженерия знаний. Модели и методы. учебник / Санкт-Петербург, 2016. Сер. Учебники для вузов. Специальная литература. 324с.
- [Macik et al., 2014] M.Macik, T. Cerny, P. Slavik Context-sensitive, cross-platform user interface generation//• Journal on Multimodal User Interfaces. 2014. Volume 8, Issue 2, pp 217-229.
- [Macik, 2016] M. Macik Automatic User Interface Generation//Thesis. DOI: 10.13140/RG.2.2.23963.26401. 2016
- [Белоусова и др., 2014] Белоусова С.А., Рогозов Ю.И. Анализ подходов к созданию пользовательского интерфейса // Известия Южного федерального университета. Технические науки. – 2014
- [Gribova et al., 2017] Valeria Gribova, Alexander Kleschev, Philip Moskalenko, Vadim Timchenko, Leonid Fedorischev, Elena Shalfeeva. The IACPaaS Cloud Platform: Features and Perspectives // 2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC) (Vladivostok, Russia, 25-29 sept. 2017). IEEE. 2017. P. 80-84. DOI: 10.1109/RPC.2017.8168076.
- [Gribova et al., 2015] Gribova V., Kleschev A., Krylov D., Moskalenko P., Timchenko V., Shalfeyeva E. A Cloud Computing Platform for Lifecycle Support of Intelligent Multi-agent Internet-services // International Conference on Power Electronics and Energy Engineering (PEEE). Hong Kong, PEOPLES R CHINA. Apr 19-20, 2015. - USA. Lancaster: Destech Publications, Inc. Pp. 231-235. ISBN 978-1-60595-256-7.



# ONTOLOGICAL APPROACH TO GENERATION OF ADAPTIVE WIMP-INTERFACES OF THE EDITORS OF KNOWLEDGE BASES

V.V. Gribova (*gribova@dvo.ru*),  
L.A. Fedorischev (*fleo1987@mail.ru*)  
Institute of automation and control processes FEB RAS,  
Vladivostok

The paper presents an approach to generating adaptive WIMP interfaces of knowledge editors on the domain ontology and knowledge base on the rules of their design. We consider ontologies of WIMP-elements, knowledge base on interface design, interface model. A short algorithm for generating the knowledge base interface is presented.

**Keywords:** user interface, ontology, base of knowledge, adaptivity, WIMP