

## ИНЖИНИРИНГ ОНТОЛОГИЙ

УДК 004.5

Научная статья

DOI: 10.18287/2223-9537-2022-12-2-200-217



### Онтологии для разработки и генерации адаптивных пользовательских интерфейсов редакторов баз знаний

© 2022, В.В. Грибова<sup>1,2</sup> ✉, С.В. Паршкова<sup>1</sup>, Л.А. Федорищев<sup>1,2</sup>

<sup>1</sup> Институт автоматизации и процессов управления ДВО РАН, Владивосток, Россия

<sup>2</sup> Владивостокский государственный университет экономики и сервиса, Владивосток, Россия

#### Аннотация

Рассматривается метод создания автоматически генерируемых адаптивных пользовательских интерфейсов редакторов баз знаний, построенных на основе онтологического подхода, с целью улучшения качества работы по формированию и редактированию баз знаний с учётом специфики предметной области, характеристик пользователя-эксперта и других параметров. Приведено описание концепции авторского подхода к генерации адаптивных интерфейсов используемых онтологий, баз знаний и моделей, как ключевых элементов предложенного подхода. В онтологии знаний о проектировании интерфейса вводятся новые абстрактные элементы, определяющие интерфейсные задачи без явного их визуального и функционального представления. Каждый абстрактный элемент этой онтологии имеет несколько вариантов представления – адаптаций абстрактных элементов. Онтология графических элементов пользовательского интерфейса определяет конкретное визуальное и функциональное представление абстрактных элементов в виде готовых повторно используемых таких элементов. База знаний о проектировании интерфейса содержит правила его формирования в зависимости от структуры онтологии предметной области, характеристик пользователя, требований удобства и простоты использования. Модель интерфейса описывает структуру сформированного адаптивного интерфейса в соответствии с внесёнными индивидуальными предпочтениями пользователя и другой дополнительной информацией. Все рассмотренные онтологии и базы знаний реализованы на платформе *IACaaS*.

**Ключевые слова:** онтология, база знаний, интерфейс, адаптивный интерфейс, генерация интерфейса, шаблоны, WIMP-интерфейс, редактор баз знаний.

**Цитирование:** Грибова В.В., Паршкова С.В., Федорищев Л.А. Онтологии для разработки и генерации адаптивных пользовательских интерфейсов редакторов баз знаний // *Онтология проектирования*. 2022. Т.12, №2(44). С.200-217. DOI:10.18287/2223-9537-2022-12-2-200-217.

**Финансирование:** работа поддержана Российским фондом фундаментальных исследований (проект 20-07-00670 А).

**Конфликт интересов:** авторы заявляют об отсутствии конфликта интересов.

#### Введение

Интеллектуальные системы с базами знаний (БЗ): экспертные системы, системы поддержки принятия решений, компьютерные тренажёры и др., широко используются во многих предметных областях (ПрО): медицина, экономика, химия и биология, военное дело, инженерное дело, экология, производство, управление процессами, юриспруденция и др. Именно БЗ является «бутылочным горлышком» таких систем [1-4]. Основная сложность связана с их формированием и сопровождением экспертами ПрО. Для упрощения создания и сопровождения БЗ активно применяется онтологический подход, который в настоящее время является стандартом для их создания (см. также [5, 6]). Вместе с тем, наличие онтологии, по которой

формируется БЗ, недостаточно для удобной работы экспертов ПрО. Немаловажное значение имеют также онтологическая модель знаний и пользовательский интерфейс, который является ключевым фактором при принятии решения об использовании программного средства.

Для работы с онтологиями и БЗ применяются различные онтологические редакторы знаний (*IWE, Protégé, OntoEdit, GrOWL, Graphl, RDFGravity, WebVOWL, Ontolingua* и др.). Пользовательские интерфейсы этих редакторов реализуют фиксированный сценарий без возможности настройки и адаптации к потребностям пользователей (некоторые редакторы имеют режимы выбора просмотра информации, но не позволяют редактирование). Объёмы БЗ для интеллектуальных систем включают тысячи и десятки тысяч понятий на разных уровнях иерархии, что является дополнительным фактором, усложняющим использование таких редакторов экспертами ПрО. Актуальной задачей является разработка методов создания редакторов БЗ с адаптивным пользовательским интерфейсом.

Проблеме адаптации пользовательских интерфейсов посвящено большое количество работ. Предложены онтологии для динамического моделирования пользователей, профили которых в дальнейшем могли использоваться для адаптивного поиска требуемой информации и неявного сбора информации с помощью фоновых сервисов (см. например, [7, 8]). В данных работах рассматривается адаптивный поиск с учётом индивидуальных параметров пользователя, но не затрагиваются вопросы выбора интерфейсных элементов и отображения информации, порядка её вывода и т.д.

Предложены различные схемы и алгоритмы разделения интерфейса по уровням абстракции и генерации адаптированного интерфейса на основе онтологий и других элементов [3, 9-12]. В данных работах можно отметить существенный прогресс в адаптивных технологиях интерфейса: рассмотрены различные концепции адаптивных систем для различных устройств и для генерации интерфейса, в том числе в зависимости от меняющихся условий внешней среды. Однако по-прежнему отсутствует связь адаптации интерфейса непосредственно с ПрО и пользователем.

В работах [13, 14] предложены достаточно полноценные концепции адаптивного пользовательского *WIMP*<sup>1</sup>-интерфейса, в том числе так называемые «пластичные» пользовательские интерфейсы. Данные работы посвящены узким задачам построения интерфейса и не решают проблему адаптивного интерфейса на основе онтологии произвольной ПрО.

Общий недостаток отмеченных подходов состоит в том, что они не учитывают в своей структуре онтологии знания, с которыми будет работать интерфейс. Поэтому для улучшения качества работы по формированию и редактированию БЗ на основе онтологий актуальны исследования и разработки в области автоматической генерации адаптивного интерфейса редактора знаний, строящегося с учётом специфики ПрО, характеристик пользователя-эксперта и других параметров, прежде всего, требований удобства использования.

Целью данной работы является детальное описание авторского подхода к разработке адаптивных *WIMP*-интерфейсов на основе онтологии ПрО и онтологической БЗ о проектировании интерфейсов.

## 1 Концепция разработки *WIMP*-интерфейсов для редакторов баз знаний

В работах [15, 16] описан подход к автоматической генерации редакторов знаний, основная идея которого заключается в следующем: на основе знаний о проектировании интерфейса, онтологии ПрО и модели пользователя автоматически генерируется пользовательский интерфейс редактора знаний (см. рисунок 1).

<sup>1</sup> WIMP - «Windows, Icons, Menus, Pointers» — окна, значки, меню, указатели.

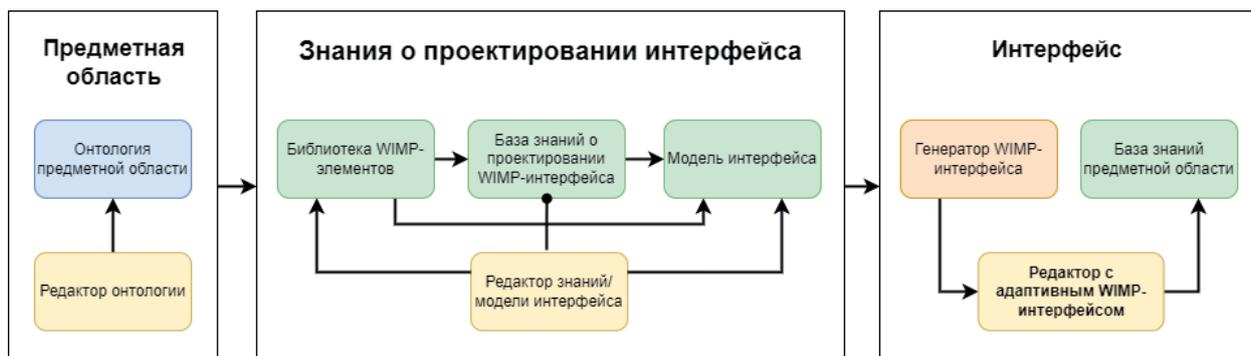


Рисунок 1 – Формирование редактора с адаптивным WIMP-интерфейсом

Реализация подхода включает следующие работы.

- 1) *Введение абстрактных элементов интерфейса.* Они определяют интерфейсные задачи без явного определения их визуального и функционального представления. Каждый абстрактный элемент интерфейса имеет несколько вариантов представления – *адаптаций абстрактных элементов*. Выбор конкретной адаптации определяется в зависимости от выполнения ряда условий (операционной системы, устройства, структуры информации, типа пользователя, его предпочтений, среды использования и др.). Абстрактные элементы интерфейса используются для описания знаний о проектировании интерфейса: для каждой интерфейсной задачи (абстрактного элемента интерфейса) и множества условий описываются возможные способы их отображения в WIMP-интерфейсе.
- 2) *Формирование базы WIMP-элементов на основе онтологии WIMP-элементов интерфейса.* Каждая адаптация абстрактного элемента имеет конкретное визуальное и функциональное представление в виде готовых повторно используемых WIMP-элементов, таких как *списки, кнопки, меню и т.д.*
- 3) *Формирование БЗ о проектировании WIMP-интерфейса.* БЗ содержит правила его формирования в зависимости от структуры онтологии ПрО, характеристик пользователя, требований удобства использования.
- 4) *Построение модели интерфейса,* которая определяет иерархию, расположение и порядок отображения интерфейсных элементов в процессе формирования БЗ, принимая во внимание предпочтения пользователя: интерфейсный стиль, тип диалога, уровень владения компьютером и другую информацию о нём.
- 5) *Разработка генератора интерфейса,* который по онтологии ПрО, БЗ о проектировании интерфейса и модели интерфейса формирует пользовательский интерфейс редактора знаний, ориентированного на конкретного пользователя.

Для реализации предложенного подхода авторами разработаны следующие онтологии:

- онтология БЗ о проектировании WIMP-интерфейса;
- онтология WIMP-элементов;
- онтология модели интерфейса.

Для формального представления онтологий в работе [17] предложен метаязык, который в течение ряда лет используется на платформе IASPaas. Онтология представляется иерархическим графом, вершинами графа являются термины онтологии, а дуги определяют следующие типы отношений для описания правил порождения знаний на основе онтологии:

- *~copy* — «копия», термин онтологии переносится без изменений в БЗ;
- *~one* — «в точности один», можно породить только один элемент в БЗ, соответствующий термину онтологии;
- *~set* — «непустое множество», возможно порождение одного или нескольких отличающихся по имени элементов по соответствующему термину онтологии;

- *~seq* — «непустая последовательность», порождение элементов в БЗ с автоматически сгенерированной нумерацией (порядком);
- *~сорутт* — «возможное отсутствие», аналог «*~сору*», но порождение не является обязательным для полноты базы;
- *~онетт* — «ноль или один», аналог «*~опа*», но порождение не является обязательным для полноты базы;
- *~сеттт* — «возможно пустое множество», аналог «*~сет*», но порождение даже одного элемента не является обязательным для полноты базы;
- *~сеттм* — «возможно пустая последовательность», аналог «*~ seq*», но порождение даже одного элемента последовательности не является обязательным для полноты базы;
- *~проху* — «прокси», служит для организации порождаемых элементов в условные группы с «невидимым» контейнером.

## 2 Онтология знаний о проектировании интерфейса

Онтология знаний о проектировании интерфейса определяет соответствия между абстрактными интерфейсными элементами и их возможными представлениями в интерфейсе в зависимости от ряда условий.

*Абстрактными интерфейсными элементами (задачами)* являются:

- элемент ввода – элемент, предоставляющий возможность ввести данные через подходящий интерфейсный компонент;
- элемент вывода – элемент, обеспечивающий возможность вывести данные через подходящий интерфейсный компонент;
- выбор – элемент для выбора элементов из множества, которое задаётся либо явно, через указание всех элементов, либо генерируется автоматически;
- контейнер – интерфейсный элемент для объединения нескольких элементов интерфейса в рамках одной задачи.

Каждая абстрактная задача имеет свои подзадачи. Например, задача выбора имеет подзадачи: выбор одного варианта из множества, выбор подмножества вариантов из множества.

Можно рассмотреть онтологию абстрактных интерфейсных элементов:

```

~сору Абстрактные интерфейсные элементы {
  ~сору Выбор {<Уникальные подзадачи>}
  ~сору Ввод {<Уникальные подзадачи>}
  ~сору Вывод {<Уникальные подзадачи>}
  ~сору Контейнер {<Уникальные подзадачи>}
}
<Уникальные подзадачи>:: { Уникальная подзадачаi}2
~сору<Уникальная подзадачаi>{
  ~сору<Реализация по умолчанию для задачи >
  ~сору Элементы интерфейса {
    ~сет Элемент интерфейса {
      ~сору<Данные, влияющие на выбор элемента интерфейса>
      ~сору<Реализация элемента интерфейса>
    }
  }
}

```

*Элементы интерфейса* – это множество вариантов реализации указанной подзадачи - элементов интерфейсов. Выбор того или иного элемента зависит от условий, которые прописаны внутри каждого элемента.

<sup>2</sup> Каждая из подзадач имеет идентичную структуру.

Реализация по умолчанию для задачи – это базовое представление абстрактного элемента интерфейса на основе заданного по умолчанию WIMP-элемента с конкретными параметрами.

Данные, влияющие на выбор элемента интерфейса, содержат описание параметров входных данных (элементов знаний), согласно которым выбирается указанный элемент интерфейса для реализации выбранной подзадачи.

```

~сору Данные, влияющие на выбор элемента интерфейса {
  ~сорулт Онтологические конструкции { ... }
  ~сорулт Особенности данных {
    ~set Особенность [string]
  }
  ~опетт Количество потомков [integer]
  ~опетт Минимальное количество потомков [integer]
  ~опетт Максимальное количество потомков [integer]
}

```

Онтологические конструкции – описание, содержащее множество условий для определения элемента интерфейса для данного элемента онтологии.

```

~сорулт Онтологические конструкции {
  ~сорулт Конструкция {
    ~сору Элементы онтологии {
      ~сорулт Список или альтернатива {СПИСОК | АЛЬТЕРНАТИВА}
      ~сору Тип элемента { нетерминал | терминал-сорт | терминал-значение}
      ~сорулт Тип значения {Строковое | Целое | Вещественное | Логическое | Дата и время | Бинарные значения}
      ~сору Спецификатор (тип отношения) { сору | сорулт | one | onett | list | listmm | set | setmm | seq | seqmm | проху}
    }
    ~сорулт Уровень размещения (глубина вложенности) {I | >I}
    ~сорулт Элементы онтологии, расположенные на одном уровне {
      ~set->Элементы онтологии { ... }
      ~сорулт Элементы онтологии отсутствуют [string]
      ~seqлт Логическая операция И | ИЛИ | НЕ}
    }
    ~сорулт Содержимое {
      ~set->Элемент { ... }
      ~seqлт Логическая операция {И | ИЛИ | НЕ}
    }
    ~сорулт Количество потомков {
      ~опетт Фиксированное количество потомков
      ~опетт Минимальное количество потомков
      ~опетт Максимальное количество потомков
    }
  }
}
}

```

Каждая конструкция содержит: элементы онтологии; уровень размещения; элементы онтологии, расположенные на одном уровне; содержимое; количество потомков.

Элементы онтологии – раздел, который содержит множество элементов онтологии (представленных в виде перечня свойств), которые необходимо преобразовать в элемент интерфейса.

Уровень размещения – указание, на каком уровне в онтологии размещён элемент (по иерархии). Было выявлено, что есть разница в реализации для некоторых элементов, которые могут находиться на первом уровне размещения в онтологии.

Элементы онтологии, расположенные на одном уровне, – список из элементов онтологии, которые могут находиться на одном иерархическом уровне с текущим элементом онтологии.

*Содержимое* - дочерние элементы для текущего элемента онтологии. Может содержать логическую операцию между элементами.

*Количество потомков* – описывает границы возможного количества потомков.

Конструкции и их элементы описаны в онтологии, представленной на рисунке 2.

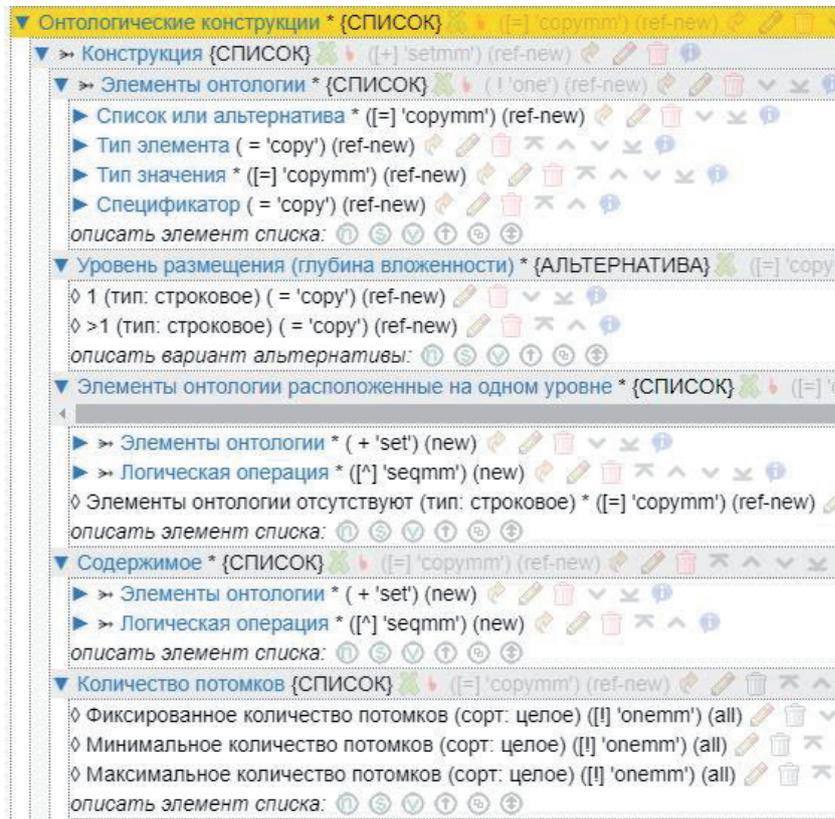


Рисунок 2 – Скриншот раздела «онтологические конструкции» на платформе IASPaas

*Особенности данных (знаний)* – элемент онтологии, содержащий условия, которые определяют интерфейсный элемент для реализации в зависимости от структуры входных данных (знаний).

*Количество потомков* определяет условие по числу дочерних элементов. Например, если у элемента «список-выбор» меньше 5 дочерних элементов, то его можно представить в виде группы радиокнопок, если же больше 5, то лучше использовать выпадающий список (см. рисунок 3).

*Реализации элемента интерфейса* (см. рисунок 4) состоят из множества *Реализаций*. Каждая *Реализация* состоит из двух частей: *Характеристики устройств* (для которых подходит данная реализация) и *Дизайн* – непосредственное определение интерфейсного отображения элемента. Дизайн определяется с помощью элементов *CommonElements* либо через уникальный набор *CSS*-свойств<sup>3</sup>. *CommonElements* – один из *WIMP*-элементов БЗ для применения в данной реализации. *CSS* – описание стилей *WIMP*-элементов: {ширина (элемента), высота, цвет фона, размер текста и др. по аналогии с каскадными таблицами стилей из веб-технологий [18]}.

Онтология *Реализации элемента интерфейса*.

~copy Реализации элемента интерфейса {

~set Реализация {

<sup>3</sup> CSS - Cascading Style Sheets, каскадные таблицы стилей.

```

~copy Характеристики устройств {
    ~set Устройство { ... }
}
~copy Дизайн {
    ~copy->CommonElements { ... }
    ~copy->CSS{ ... }
}
}
}
    
```

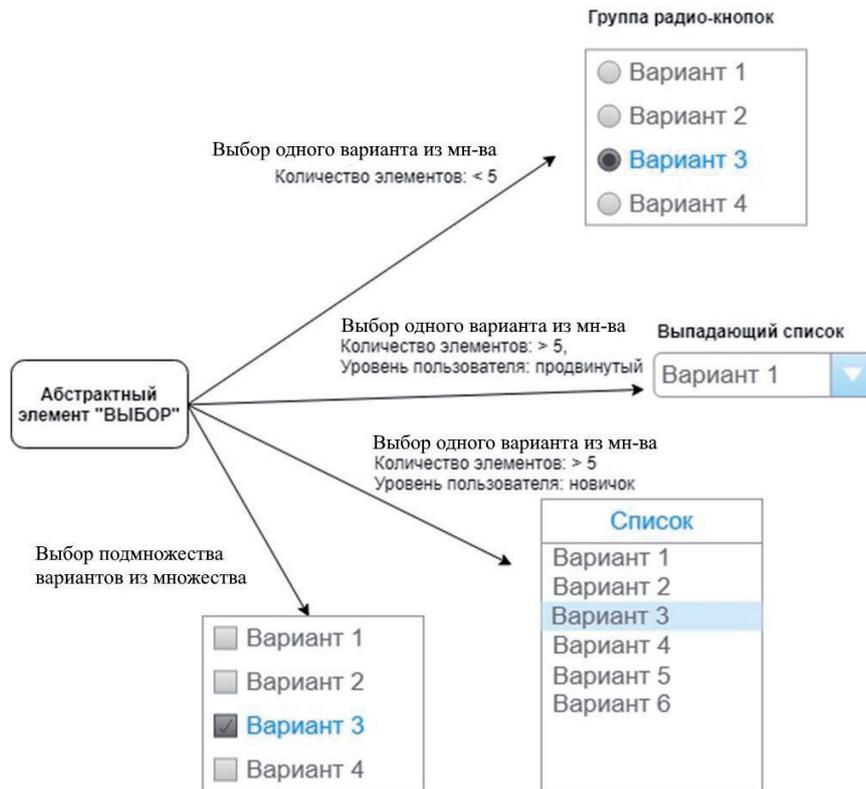


Рисунок 3 – Схема реализации условий абстрактным элементом «Выбор» в редакторе с адаптивным WIMP-интерфейсом

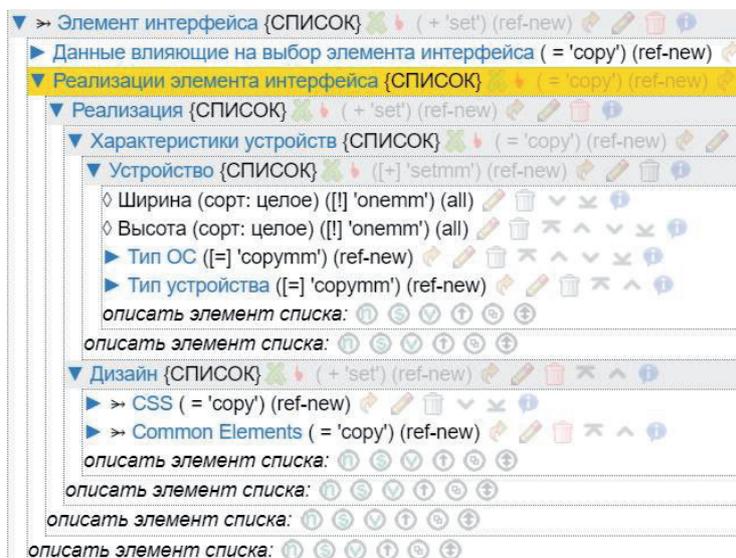


Рисунок 4 – Скриншот раздела онтологии «Реализации элемента интерфейса» на платформе IACPaas

На основе описанной онтологии сформирована БЗ о проектировании интерфейса. Размер базы – 1812 понятий, число отношений – 1864.

На рисунке 5 представлен пример описания абстрактной задачи в БЗ на основе структуры <Выбор одного варианта из множества> в задаче «Выбор»:

```

~сору Выбор {
  ~сору Выбор одного варианта из множества {
    ~сору<Реализация по умолчанию для задачи >
    ~сору Элементы интерфейса {
      ~set Элемент интерфейса {
        ~сору<Данные, влияющие на выбор элемента интерфейса>
        ~сору<Реализация элемента интерфейса>
      }
    }
  }
}

```

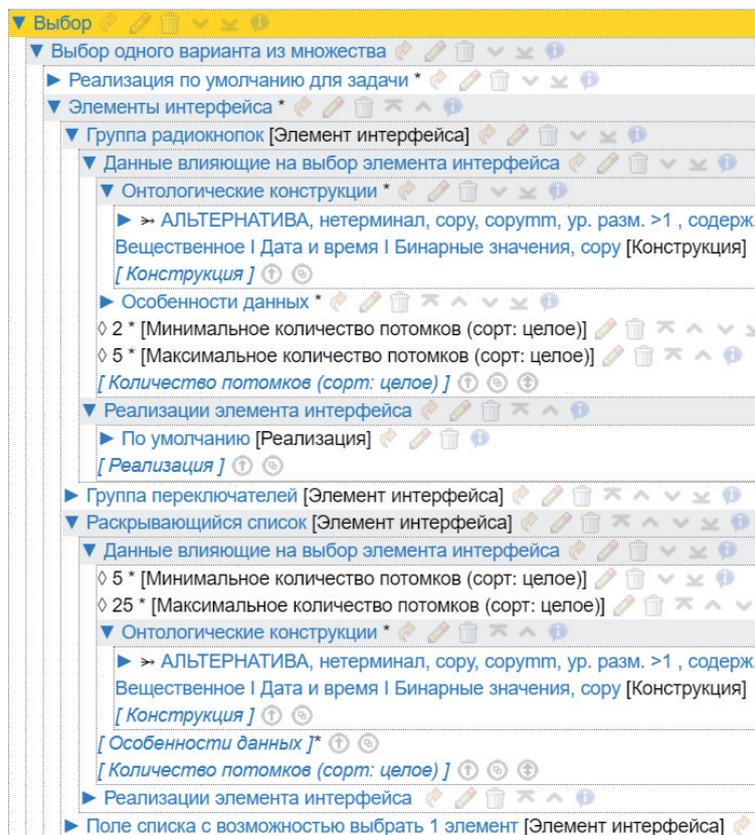


Рисунок 5 – Скриншот фрагмента БЗ о проектировании интерфейса на платформе IACPaas

Иллюстрация адаптивной генерации интерфейсного элемента на основе сформированной БЗ абстрактных элементов по представленной выше онтологии показана на рисунке 6 на примере генерации редактора БЗ по онтологии ПроО.

В представленной на рисунке схеме видно, что для абстрактного элемента «Выбор» заданы несколько возможных вариантов элементов интерфейса: «Группа радиокнопок», «Группа переключателей», «Раскрывающийся список» и др. Для каждого из этих интерфейсных элементов выбора заданы онтологические конструкции, для которых данный элемент может подходить. Например, для группы радиокнопок указаны онтологические конструкции: «Альтернатива, нетерминал, сору, сорутт,...». Такие же конструкции заданы и для элемента «Раскрывающийся список». Каждый из этих двух вариантов будет зависеть от

«Данных, влияющих на выбор элемента интерфейса». В первом случае заданы параметры количества элементов от 2 до 5, во втором случае от 5 до 25. Справа на схеме видно, что представленная онтология ПрО соответствует данным элементам интерфейса. Пример БЗ 1 по этой онтологии генерирует в результате группу радиокнопок, Пример БЗ 2 – раскрывающийся список в соответствии с количеством элементов.

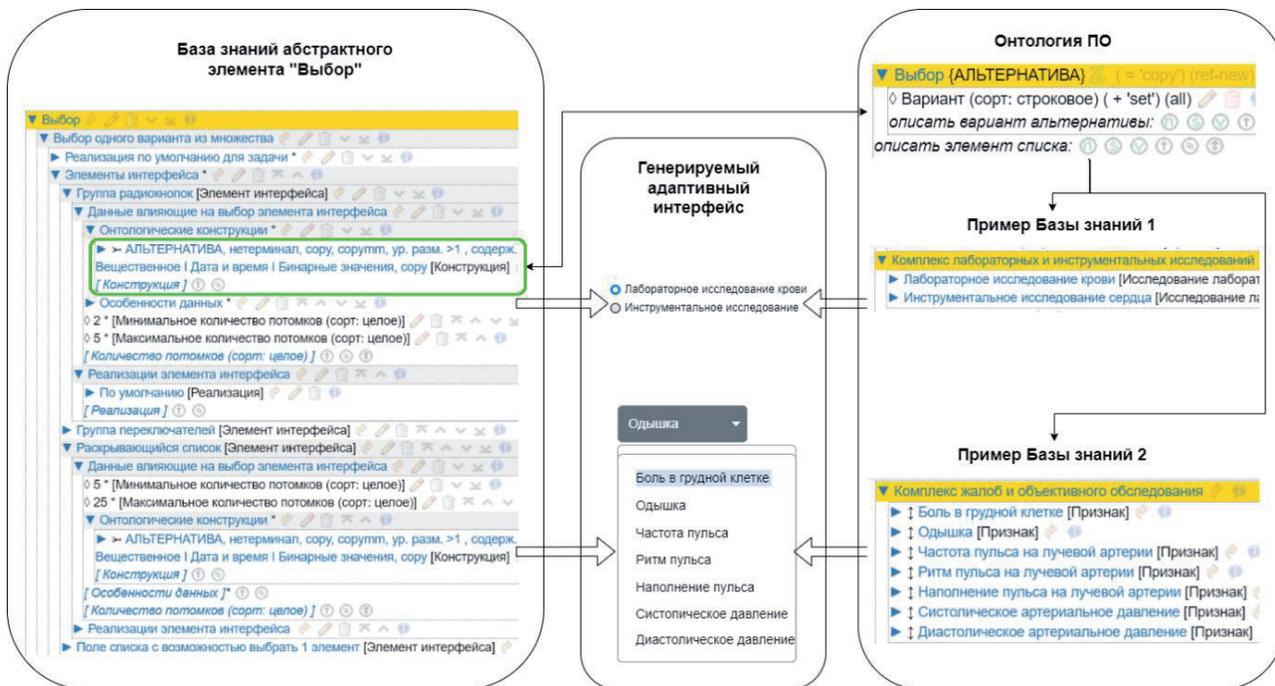


Рисунок 6 – Пример адаптивной генерации интерфейса

### 3 Онтология WIMP-элементов

Онтология WIMP-элементов состоит из двух основных разделов:

- множество WIMP-элементов (независимые интерфейсные единицы, несущие собственную смысловую и функциональную нагрузку: кнопка, поле ввода, список и др.);
- множество стилей CSS (стили определяют оформление внешнего вида интерфейсных элементов, начиная от простых, таких как шрифт, цвет, размер элементов, до составных, содержащих наборы интерфейсных решений).

```

Онтология WIMP элементов {
    ~copyCommonElements{...}
    ~copyCSS {...}
}
    
```

Онтология содержит 38 базовых элементов интерфейса <~copyWIMP-элементы>, среди которых представлены такие элементы как: кнопка (*Button*), чек-боксы (*CheckboxGroup*), радиокнопки (*RadioButtonGroup*), текст (*Text*), текстовое поле ввода (*Input*), комбинированное поле со списком (*Combobox*), кнопка пошаговой прокрутки (*Stepper*) и др. Структурно каждый из этих элементов в онтологии WIMP состоит из некоторого набора уникальных для данного элемента атрибутов и стилей CSS (*options*).

```

~setmm WIMP-элемент {
    ~copyoptions
    <Набор уникальных свойств>}
    
```

Элемент интерфейса *Div* является базовым универсальным элементом-контейнером с возможностью включения других элементов. Основная особенность его определения заклю-

чается в «рекурсивности»: *Div* – это *WIMP*-элемент, способный включать как другие элементы, так и элементы того же типа (*Div*). Этот элемент (также как и другие *WIMP*-элементы) содержит собственный набор стилей *CSS*.

```

~set Div: {
  ~set ->Div
  ~set ->CSS
  ~set ->Text
  <... другие элементы>
}
~setmm Text: {
  ~copy->options
  ~onetext
}

```

Кнопка (*Button*) является сложным элементом и состоит из группы более простых элементов. Кнопка всегда содержит оформление блока *CSS*, а так же может содержать текст (*Text*) и пиктограмму (*Ico*), располагаемую на кнопке; подсказку (*Hint*), которая выводится на экран при наведении курсора на кнопку. Помимо этого кнопка содержит раздел, отвечающий за состояния (*States*), в которых она может находиться.

```

~setmm Button: {
  ~copy->options
  ~copymm -> Text
  ~copymm -> Hint
  ~copymm ->Ico
  ~copy states { ~set ->options }
}

```

Текстовое поле ввода (*Input*) содержит текст (*Text*) рядом с полем, описывающий данные, которые необходимо ввести. Возможно присутствие элемента (*Div*), отвечающего за токенизатор ввода, и наличие текста внутри поля (*Text*), подсказывающего возможное вводимое значение.

```

~setmm Input: {
  ~copy->options
  ~copy -> Text
  ~copymm ->Text
  ~copymm ->Div
}

```

*WIMP*-элементы *Combobox* и *RadioGroup* характерны тем, что определяют не один элемент, а несколько. Элементы *Combobox* и *RadioGroup* в большей степени определяют форму и поведение группы. Каждый из них содержит ссылку на соответствующий элемент и текст (*Text*), отвечающий за заголовок группы.

```

~setmmRadioButtonGroup: {
  ~copy -> Checkbox
  ~copy -> Text
  ~copy->options
}
~setmmCheckboxGroup: {
  ~copy ->Radiobutton
  ~copy ->Text
  ~copy->options
}

```

На платформе *IACPaaS* разработана онтология *WIMP* элементов, соответствующая приведённому формальному описанию (см. рисунок 7).

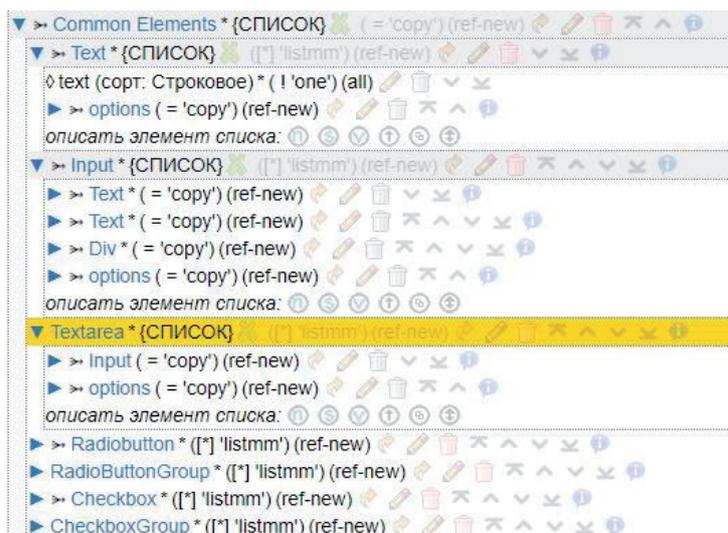


Рисунок 7 – Скриншот фрагмента онтологии WIMP элементов

Стили CSS определяются в онтологии по принципу максимальной аналогии и сходства с традиционными стилями CSS, известными в веб-разработке. Стиль CSS определяется парой  $\langle selector, options \rangle$ :

```
~copyCSS {
  ~copyselector {<...>}
  ~copyoptions {<...>}
}
```

Селектор (*selector*) так же, как и в традиционных CSS, определяет, для каких именно элементов интерфейса применить заданные правила (свойства) и в онтологии определяется так:

```
~copy selector: {
  ~set tag: <*, Base, Text, Button, Div, List, ...>
  ~set class: String
  ~set id: String
}
```

Тело (*body*) стиля  $\langle \sim copyoptions \rangle$  содержит набор свойств, которые будут применяться для соответствующих селектору элементов. Набор свойств стилей определяется по аналогии с традиционными CSS из веб-разработки: ширина, высота, цвет, отступы и т.д.

```
~copy body: {
  ~one width
  ~one height
  ~one layout: [vertical, horizontal, frame (поверх всех окон), ...]
  ...
}
```

На рисунке 8 представлено описание фрагмента онтология CSS-стилей.

#### 4 Онтология модели интерфейса

Онтология модели интерфейса состоит из двух основных компонентов: {Сохраненные интерфейсы, Информация о пользователе}.

```
Онтология модели интерфейса {
  ~copy Сохраненные интерфейсы {...}
  ~copy Информация о пользователе {...}
}
```

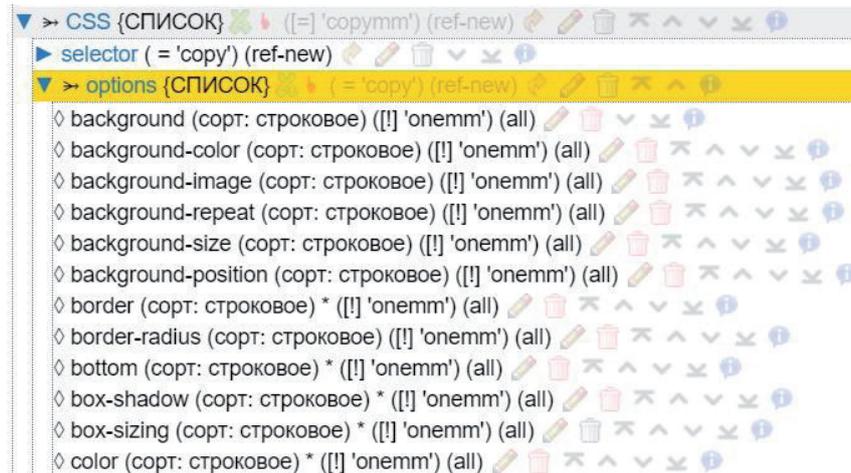


Рисунок 8 – Скриншот фрагмента онтологии CSS-стилей

Элемент онтологии «Сохраненные интерфейсы» определяет структуру для сохранения и отображения интерфейса, специфичного для каждого пользователя. Данная структура позволяет запоминать все выбранные настройки и изменения в интерфейсе, сделанные для конкретного пользователя. Индивидуальный интерфейс пользователя состоит из 5 основных элементов: {Размещение элементов, Внешний вид элементов, Шаблон, Вычисляемая информация о поведении пользователя, Общий CSS}.

```

Сохраненные интерфейсы {
  ~set Имя сервиса {
    ~set Размещение элементов
    ~set Внешний вид элементов
    ~coru Шаблон
    ~coru Вычисляемая информация о поведении пользователя
    ~coru Общий CSS
  }
}

```

«Размещение элементов» определяет структуру записи изменений для элементов ПрО, позиции которых были изменены в результате пользовательского редактирования модели интерфейса. «Размещение элементов» - это множество вариантов размещения отдельных элементов онтологии. «Размещение элементов» содержит список «Поэлементная информация», включающий «Элемент онтологии» и «Позиционирование элемента интерфейса»

```

Размещение элементов {
  ~set Поэлементная информация {
    ~coru Элемент онтологии
    ~coru Позиционирование элемента интерфейса
  }
}

```

«Шаблон» определяет ссылку на один из шаблонов, указанных в онтологии о проектировании интерфейса, он может дополняться новым шаблонами. «Шаблон» – это указатель на то, при каких шаблонах интерфейса может быть выбран и реализован данный элемент интерфейса. У всех элементов есть шаблон по умолчанию. Онтология шаблонов:

```

Шаблоны {
  ~set Шаблон {
    ~set -> Элемент интерфейса {...}
    ~coru -> Информация о шаблоне {...}
  }
}

```

На рисунке 9 представлен фрагмент онтологии модели интерфейса на платформе IACPaas.

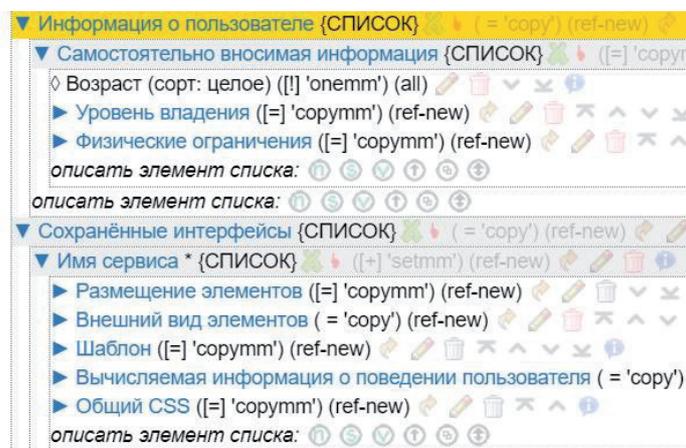


Рисунок 9 – Фрагмент онтологии модели интерфейса

## 5 Онтология модели пользователя

Для генерации окончательного вида интерфейса большое значение имеют данные, индивидуально характеризующие пользователя интерфейса. Для этого предусмотрена онтология модели пользователя, в которой предусмотрены специальные критерии, помогающие выделить индивидуальные параметры пользователя и содержащиеся в информации о пользователе. «Модель пользователей» включает: {Характеристики пользователей, Вычисляемые характеристики в процессе работы, Особенности реализации, Источники}.

```

Модели пользователей {
  ~set Модель {
    ~сору Характеристики пользователей {...}
    ~сору Вычисляемые характеристики в процессе работы {...}
    ~сору Особенности реализации {...}
    ~сору Источники {...}
  }
}
    
```

«Характеристики пользователей» включают данные о пользователе: {Возраст, Уровень владения, Физические ограничения}.

```

Характеристики пользователей {
  ~set Пользователь {
    ~сору Возраст {...}
    ~сору Уровень владения {...}
    ~сору Физические ограничения {...}
  }
}
    
```

«Особенности реализации» содержат «Нерекомендуемые элементы» к использованию в рамках данной модели и «Рекомендуемые элементы».

```

Особенности реализации {
  ~сорутт Нерекомендуемые элементы
  ~сору Рекомендуемые элементы
}
    
```

«Нерекомендуемые элементы» содержат список элементов интерфейса, которые не рекомендуется использовать в модели. Приоритет генерации у элементов данного списка меньше, чем у других элементов. Каждый такой элемент интерфейса описывается по аналогии с обычным элементом интерфейса, представленным ранее:

```

Нерекомендуемые элементы {
  ~set -> Элемент интерфейса
}
    
```

«Рекомендуемые элементы» определяют набор стилей и общих элементов, которые будут присущи данной модели интерфейса:

```

Рекомендуемые элементы {
  ~copumt -> Common Elements
  ~copy -> CSS
}
    
```

*Common Elements* и *CSS* определяются так же как в онтологии базы *WIMP*-элементов (см. раздел 3). На рисунке 10 представлен фрагмент онтологии «Модель пользователя», реализованный на платформе *IACPaas*.

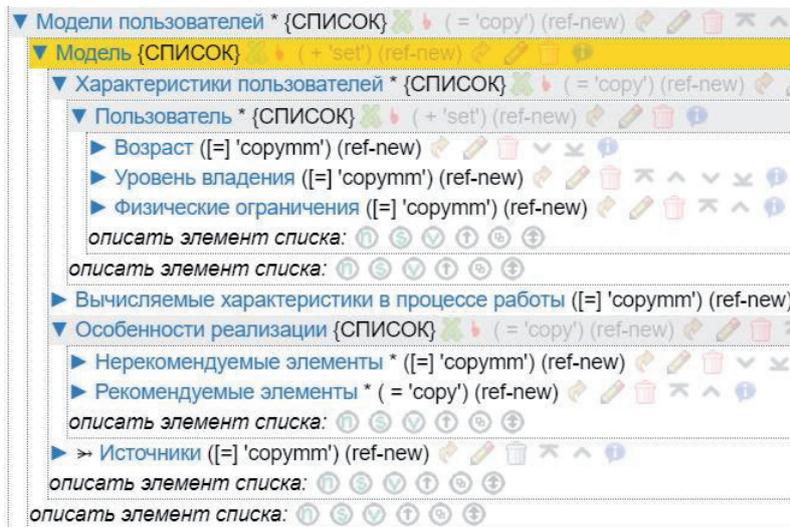


Рисунок 10 – Фрагмент онтологии «Модель пользователя»

Упрощённый пример генерации адаптивного интерфейса с использованием рассмотренных онтологий иллюстрирует рисунок 11. В представленной на рисунке 11 схеме строятся два варианта интерфейса: на основе примера онтологии ПрО (онтологии диагностики) и адаптации по устройству (ПК или смартфон).

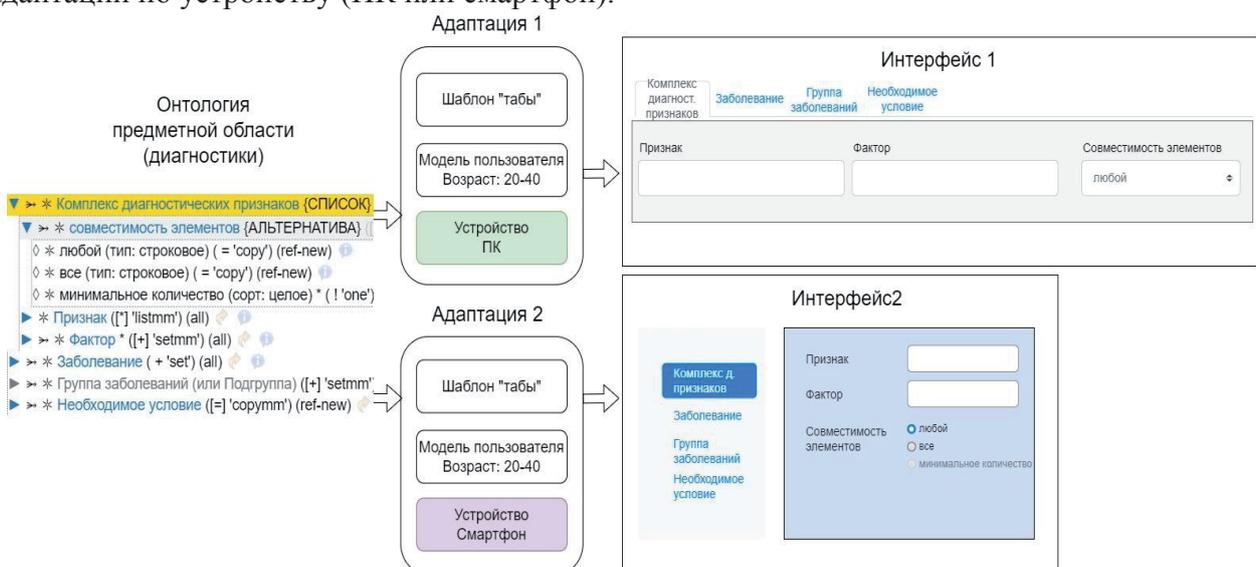


Рисунок 11 – Пример генерации адаптивного интерфейса

## Заключение

Разработка интерфейсов редакторов БЗ является актуальной задачей в инженерии знаний. Это вытекает из требований современного этапа к разработке систем на основе знаний: БЗ должны разрабатывать и сопровождать эксперты ПрО. Учитывая разный уровень знаний экспертов в области информационных технологий, удобные и понятные редакторы знаний являются определяющим фактором включения экспертов в процесс разработки БЗ. Для автоматической генерации интерфейса редакторов БЗ используется онтологический подход, включающий разработку БЗ о проектировании *WIMP*-интерфейса, библиотеки *WIMP*-элементов, модели интерфейса. Разработанный генератор интерфейса автоматически формирует пользовательский интерфейс редактора знаний, ориентированный на конкретного пользователя.

В настоящее время на платформе *IACPaas* реализованы все информационные и программные компоненты, проводится их опытная эксплуатация.

## Список источников

- [1] **Щеглов С.Н.** Онтологический подход и его использование в системах представления знаний // Известия Южного федерального университета. 2009. №4(93). С.146-153.
- [2] **Подлипский О.К.** Построение баз знаний группой экспертов // Компьютерные исследования и моделирование 2010. Т.2, №1, С.3-11.
- [3] **Белоусова С.А., Rogozov Ю.И.** Анализ подходов к созданию пользовательского интерфейса // Известия Южного федерального университета. Технические науки. 2014. № 6(155). С.142-148.
- [4] **Корончик Д.Н.** Пользовательские интерфейсы интеллектуальных систем // Кибернетика и программирование. 2012. № 1. С.16-22. DOI:10.7256/2306-4196.2012.1.13861.
- [5] ГОСТ Р ИСО/МЭК 21838-1-2021. Национальный стандарт Российской Федерации. Информационные технологии. Онтологии высшего уровня (TLO). Часть 1. Требования. Information technology. Top-level ontologies (TLO). Part 1. Requirements. Дата введения 2022-04-30.
- [6] ГОСТ Р 59798-2021. Национальный стандарт Российской Федерации. Информационные технологии. Онтологии высшего уровня (TLO). Часть 2. Базисная формальная онтология (BFO). Information technology. Top-level ontologies (TLO). Part 1. Basic Formal Ontology (BFO). Дата введения 2022-04-30.
- [7] **Gauch, S., Chaffee, J., & Pretschner, A.** Ontology-based personalized search and browsing // Web Intelligence and Agent Systems. 2003. P.219–234.
- [8] **Razmerita, L., Angehrn, A., & Maedche, A.** Ontology-based user modeling for knowledge management systems. *In User modeling*. Berlin, Germany: Springer. 2003. P.213–217.
- [9] **Трезубов А.С.** Разработка адаптивных контекстозависимых интерфейсов с использованием онтологических моделей // Кибернетика и программирование. 2017. № 6. С.50-56. DOI:10.25136/2306-4196.2017.6.24747.
- [10] **McAvoy, L. M., Chen, L., & Donnelly, M.** An ontology based context management system for smart environments // UBICOMM. 2012, the Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. P.18–23.
- [11] **Liu B., Chen H., He W.** Deriving User Interface from Ontologies: A Model-based Approach // 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Hong Kong China. 2005. P.254–259.
- [12] **Бубарева О.А., Вайцель Н.С.** Подход к проектированию пользовательского интерфейса в системах реального времени на базе онтологий // Южно-сибирский научный вестник. 2018. С.82-86.
- [13] **Алфимцев А.Н., Локтев Д.А., Локтев А.А.** Разработка пользовательского интерфейса комплексной системы видеомониторинга // Информационные системы и логистика в строительстве. 2012. С.242-251.
- [14] **Abdelkrim Chebieb, Yamine Ait-Ameur.** A formal model for plastic human computer interfaces // *Frontiers of Computer Science*, Springer Verlag, 2018, 12 (2). P.351-375. DOI:10.1007/s11704-016-5460-3.
- [15] **Грибова В.В., Федорищев Л.А.** Адаптивный генератор *WIMP*-интерфейса редакторов базы знаний на основе онтологии // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2019. №49. С.110-119.
- [16] **Грибова В.В., Федорищев Л.А.** Онтологический подход к генерации адаптивных *WIMP*-интерфейсов редакторов баз знаний // В сборнике: Шестнадцатая Национальная конференция по искусственному интеллекту с международным участием КИИ-2018. Труды конференции: в 2-х томах. 2018. С.19-26.

- [17] **Грибова В.В., Клецев 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.
- [18] **Мейер Э.А.** CSS. Каскадные таблицы стилей. Подробное руководство. Пер. с англ. СПб: Символ-Плюс, 2008. 576 с.

## Сведения об авторах



**Грибова Валерия Викторовна**, 1965 г.р. Окончила Ленинградский политехнический институт (1989), д.т.н. (2007), член-корр. РАН (2022). Заместитель директора по научной работе, руководитель лаборатории интеллектуальных систем Института автоматизации и процессов управления ДВО РАН, профессор Дальневосточного федерального университета (ДФУ), Владивостокского государственного университета экономики и сервиса (ВГУЭС). Вице-президент Российской ассоциации искусственного интеллекта. В списке научных трудов более 300 работ. ORCID: 0000-0001-9393-351X; AuthorID (РИНЦ): 7400; AuthorID (Scopus): 7801667631. [gribova@iacp.dvo.ru](mailto:gribova@iacp.dvo.ru). ✉.

**Паршкова Светлана Владимировна**, 1999 г.р. Окончила ВГУЭС (2020). Инженер лаборатории интеллектуальных систем Института автоматизации и процессов управления ДВО РАН. [parshkovas@iacp.dvo.ru](mailto:parshkovas@iacp.dvo.ru).



**Федорищев Леонид Александрович**, 1987 г.р. Окончил ДФУ в 2009 г., к.т.н. (2013). Старший научный сотрудник лаборатории интеллектуальных систем Института автоматизации и процессов управления ДВО РАН, доцент ВГУЭС. В списке научных трудов более 24 работ. ORCID: 0000-0002-2049-2570; AuthorID (РИНЦ): 877296; AuthorID (Scopus): 55336913500. [fleo1987@mail.ru](mailto:fleo1987@mail.ru).



Поступила в редакцию 30.05.2022, после рецензирования 02.06.2022. Принята к публикации 20.06.2022.



Scientific article

DOI: 10.18287/2223-9537-2022-12-2-200-217

## Ontologies for development and generation adaptive user interfaces of knowledge base editors

© 2022, V.V. Gribova<sup>1,2</sup>✉, S.V. Parshkova<sup>1</sup>, L.A. Fedorischev<sup>1,2</sup>

<sup>1</sup> Institute for Automation and Control Processes FEB RAS, Vladivostok, Russia

<sup>2</sup> Vladivostok State University of Economics and Service, Vladivostok, Russia

### Abstract

The article discusses a method for creating automatically generated adaptive user interfaces of knowledge base editors based on an ontological approach. The main goal that guided the authors is to improve the quality and convenience of creating and editing knowledge bases, taking into account the specifics of the subject area, the characteristics of an expert user and other parameters. The description of the concept of a new approach to generating adaptive interfaces is given. Additionally, a detailed description of all ontologies, knowledge bases and models used in the specified approach to generating interfaces, as key elements of the proposed idea. The presented ontology of knowledge about interface design introduces new abstract elements that define interface tasks without explicitly defining their visual and functional representation, while each abstract element of this ontology has several representation options - adaptations of abstract elements. The ontology of graphical elements of the user interface defines a specific visual and functional representation of abstract elements in the form of ready-made reusable WIMP elements, such as lists, buttons, menus, etc. The presented knowledge base on interface design describes the rules for its formation depending on the structure of the domain ontology, user characteristics, and usability requirements. The presented interface model describes the structure

of the generated adaptive interface in accordance with the user's individual preferences and other additional information. All the ontologies and knowledge bases described in the article are currently implemented on the IACPaaS platform.

**Key words:** *ontology, knowledge base, interface, adaptive interface, interface generation, templates, WIMP interface, knowledge base editor.*

**For citation:** *Gribova VV, Parshkova SV, Fedorishchev LA. Ontologies for developing and generating adaptive user interfaces of knowledge base editors [In Russian]. Ontology of designing. 2022; 12(2): 200-217. DOI:10.18287/2223-9537-2022-12-2-200-217.*

**Financial support:** This work was supported by the Russian Foundation for Basic Research (project 20-07-00670 A).

**Conflict of interest:** The author declares no conflict of interest.

## List of figures

- Figure 1 - Building an editor with an adaptive WIMP interface
- Figure 2 - Screenshot of the "ontological constructions" section on the IACPaaS platform
- Figure 3 - Scheme for the implementation of conditions by the abstract element "Choice" in the editor with an adaptive WIMP interface
- Figure 4 - Screenshot of the ontology section "Implementation of the interface element" on the IACPaaS platform
- Figure 5 - Screenshot of a fragment of the knowledge base on interface design on the IACPaaS platform
- Figure 6 - An example of adaptive interface generation
- Figure 7 - Screenshot of a fragment of the WIMP ontology of interface generation elements
- Figure 8 - Screenshot of CSS Styles Ontology
- Figure 9 - Fragment of the interface model ontology
- Figure 10 - Fragment of the "User model" ontology
- Figure 11 - An example of generating an adaptive interface

## References

- [1] **Shcheglov SN.** Ontological approach and its use in knowledge representation systems [In Russian]. *Izvestiya of the Southern Federal University.* 2009; 4(93): 146-153.
- [2] **Podlipsky OK.** Building knowledge bases by a group of experts [In Russian]. *Computer Research and Modeling* 2010; 2(1): 3-11.
- [3] **Belousova SA, Rogozov YuI.** Analysis of approaches to creating a user interface [In Russian]. *Izvestiya of the Southern Federal University. Technical science.* 2014; 6(155): 142-148.
- [4] **Koronchik DN.** User interfaces of intelligent systems [In Russian]. *Cybernetics and programming.* 2012; 1: 16-22. DOI:10.7256/2306-4196.2012.1.13861.
- [5] GOST R ISO/IEC 21838-1-2021. National standard of the Russian Federation. Information Technology. Top-level ontologies (TLO) [In Russian]. Part 1. Requirements. Information technology. Top-level ontologies (TLOs). Part 1. Requirements. Effective date 2022-04-30.
- [6] GOST R 59798-2021. National standard of the Russian Federation. Information Technology. Top-level ontologies (TLO) [In Russian]. Part 2. Basic formal ontology (BFO). Information technology. Top-level ontologies (TLOs). Part 1. Basic Formal Ontology (BFO). Effective date 2022-04-30.
- [7] **Gauch S, Chaffee J, Pretschner A.** Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems.* 2003. P.219–234.
- [8] **Razmerita L, Angehrn A, Maedche A.** Ontology-based user modeling for knowledge management systems. In *User modeling.* Berlin, Germany: Springer.2003. P.213–217.
- [9] **Tregubov AS.** Development of adaptive context-dependent interfaces using ontological models [In Russian]. *Cybernetics and Programming.* 2017; 6: 50-56. DOI:10.25136/2306-4196.2017.6.24747.
- [10] **McAvoy LM, Chen L, Donnelly M.** An ontology-based context management system for smart environments. *UBICOMM.* 2012, the Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, p.18–23.
- [11] **Liu B, Chen H, He W.** Deriving User Interface from Ontologies: A Model-based Approach. 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Hong Kong China. 2005. P.254–259.
- [12] **Bubareva OA, Vaytsel NS.** Ontology-Based Approach to User Interface Design in Real Time Systems [In Russian]. *South Siberian Scientific Bulletin.* 2018. P.82-86.

- [13] **Alfimtsev AN, Loktev DA, Loktev AA.** Development of a user interface for a complex video monitoring system [In Russian]. *Information systems and logistics in construction*. 2012. P.242-251.
  - [14] **Abdelkrim Chebieb, Yamine Ait-Ameur.** A formal model for plastic human computer interfaces. *Frontiers of Computer Science*, Springer Verlag, 2018; 12(2): 351-375. DOI:10.1007/s11704-016-5460-3.
  - [15] **Gribova VV, Fedorishchev LA.** Adaptive generator of WIMP-interface for knowledge base editors based on ontology [In Russian]. *Tomsk State University Bulletin. Management, computer technology and informatics*. 2019; 49: 110-119.
  - [16] **Gribova VV, Fedorishchev LA.** Ontological approach to generating adaptive WIMP interfaces for knowledge base editors [In Russian]. In the collection: *Sixteenth National Conference on Artificial Intelligence with International Participation KII-2018 Proceedings of the conference: in 2 volumes*. 2018. P.19-26.
  - [17] **Gribova VV, Kleshchev AS, Moskalenko FM, Timchenko VA.** A two-level model of complexly structured information units, corresponding to the metaphor of questioning [In Russian]. *Scientific and technical information. Ser. 2*. 2015; 10: 1-10.
  - [18] **Meyer EA.** CSS. Cascading Style Sheets. Detailed Guide [In Russian]. Plus Symbol. 2008.
- 

### About the authors

**Valeria Viktorovna Gribova** (b. 1965) graduated from the Leningrad Polytechnic Institute in 1989, D.Sc. Tech. (2007). Professor of FEFU, VSUES. Vice President of the Russian Association of Artificial Intelligence. Corresponding Member of the Academy of Engineering Sciences named after Academician A.M. Prokhorova (2016). Member of ITHEA. Expert of the Analytical Center under the Government of the Russian Federation, expert of the Russian Foundation for Basic Research, Russian Humanitarian Foundation, Russian Science Foundation, Federal State Budget Scientific Institution Research Institute RINKCE. The list of scientific works includes more than 280 works. ORCID: 0000-0001-9393-351X; Author ID (RSCI): 7400; Author ID (Scopus): 7801667631. [gribova@iacp.dvo.ru](mailto:gribova@iacp.dvo.ru) ✉.

**Svetlana Vladimirovna Parshkova** (b. 1999) graduated from VSUES in 2020. Engineer of the Laboratory of Intelligent Systems at the Institute of Automation and Control Processes FEB RAS. [parshkovas@iacp.dvo.ru](mailto:parshkovas@iacp.dvo.ru).

**Leonid Alexandrovich Fedorishchev** (b. 1987) graduated from FENU in 2009, Ph.D. (2013). Senior Researcher, Laboratory of Intelligent Systems at the Institute of Automation and Control Processes FEB RAS, associate professor of VSUES. There are more than 24 works in the list of scientific papers. ORCID: 0000-0002-2049-2570; Author ID (RSCI): 877296; Author ID (Scopus): 55336913500. [fleo1987@mail.ru](mailto:fleo1987@mail.ru).

---

Received May 30, 2022. Revised June 2, 2022, Accepted June, 20, 2022.

---