

Белорусский государственный университет информатики и радиоэлектроники,
кафедра интеллектуальных информационных технологий

П.Бровки, 6, г. Минск, Республика Беларусь
+375-017-293-80-92; e-mail: Kolb@bsuir.by

Рассматриваются средства визуализации пользовательских интерфейсов интеллектуальных систем в основе, которых лежат однородные семантические сети.

Ключевые слова – пользовательский интерфейс, семантическая сеть, интеллектуальная система, база знаний.

1 ВВЕДЕНИЕ

Человеко-машинное взаимодействие является одним из самых важных и перспективных направлений информатики. Разработчики программного обеспечения прилагают огромные усилия для поддержки адаптированных к пользователю средств общения машины и человека [1]. Однако проблема организации диалога человека и компьютерной системы до сих пор окончательно не решена.

В настоящее время существует ряд работ, которые направлены на решение проблемы интеллектуализации пользовательского интерфейса в различных стадиях жизненного цикла пользовательского интерфейса.

Подходы, связанные с интеллектуализацией процесса проектирования, описаны в работах В. В. Грибовой [2]. Данный подход предполагает унифицировать методы проектирования пользовательских интерфейсов с помощью онтологий на основе модели ориентированного подхода.

Одним из примеров подхода направленного на интеллектуализацию пользовательского интерфейса прикладной системы является подход, предложенный в работе [3]. Для интеллектуализации ряда функций пользовательского интерфейса авторы предлагают использовать экспертные системы.

Приведенные подходы снимают ряд проблем, связанных с проектированием пользовательских интерфейсов и дальнейшим их функционированием, однако не позволяют решать проблемы, возникающие при разработке пользовательских интерфейсов интеллектуальных систем. Среди таких проблем наиболее значимы следующие проблемы:

- интеграция пользовательских интерфейсов схожих систем и построение пользовательских интерфейсов на основе имеющихся компонентов,
- унификация способов представления и визуализации пользовательских интерфейсов.

Важнейшими из этих проблем, являются проблемы, связанные с унификацией визуализации и представления знаний пользовательского интерфейса.

Визуальные модели обладают особой познавательной силой, фактически представляя средства когнитивной графики для структурирования информации. На практике используется более сотни методов визуального структурирования [4], [5]. В большинстве своем в основу этих методов положены разнообразные сетевые модели представления знаний [6].

Анализируя визуальные выразительные средства современных пользовательских интерфейсов, можно выделить следующие проблемы:

- однотипность и невыразительность алфавита графических форм, используемых для визуализации элементов управления;
- отсутствие, семантических закономерностей при работе с пользовательским интерфейсом.

Существенного продвижения в решении данных проблем добились разработчики создающие интерфейсы для web-приложений и трёхмерных игр. Это обусловлено рядом причин, в первую очередь это:

- ориентация на использование библиотек для работы с графикой, не зависящих от библиотек операционной системы;
- ориентация на визуализацию близкую к образным формам предметной области;
- ориентация на графические решения, которые в первую очередь позволяют получить существенный когнитивный эффект восприятия визуализируемой информации.

Однако, не смотря на существенные преимущества подходов, используемых при разработке web-ориентированных и трёхмерных пользовательских интерфейсов, подходы, предлагаемые данными областями, не нашли распространения при разработке пользовательских интерфейсов интеллектуальных систем.

Современные тенденции развития пользовательских интерфейсов показывают, что дальнейшее развитие пользовательских интерфейсов не возможно без применения для организации интерфейса методов и средств искусственного интеллекта.

Приведем основные положения одного из подходов организации пользовательского интерфейса интеллектуальных систем, использующего методы искусственного интеллекта.

2 ОСНОВНАЯ ИДЕЯ ПОДХОДА

Научная идея предлагаемого подхода состоит в рассмотрении информационных объектов и объектов управления пользовательского интерфейса как элементов базы знаний (БЗ). Такой подход является принципиально новым и позволяет рассматривать пользовательский интерфейс как специализированную интеллектуальную систему, решающую задачу организации диалога человека и системы, обеспечивающей решение основных задач программного средства.

Для описания БЗ пользовательского интерфейса предлагается использовать однородные семантические сети с базовой теоретико-множественной интерпретацией [7]. Основным способом кодирования информации для таких сетей является SC (Semantic code)-код. Интеллектуальные системы, построенные с использованием SC-кода, называются sc-системами.

Пользовательский интерфейс каждой прикладной sc-системы визуализируется с помощью SCg-кода [7] — унифицированного способа визуализации семантических сетей закодированных с помощью SC-кода.

Минимальные, но семантически полные средства SCg-кода, обеспечивающие изображение любых конструкций SC-кода, назовем ядром SCg-кода или сокращенно SCg-ядром.

Перечислим элементы алфавита SCg-ядра:

- *sc.g-узел* трактуется как графическим изображением sc-узла неизвестного семантического типа (семантика которого дополнительно уточняется явно средствами SC-кода);

- *sc.g-ребро* трактуется как графическое изображением sc-ребра общего вида (неориентированной пары sc-элементов, семантика которой дополнительно уточняется явно средствами SC-кода);

- *обобщенная sc.g-дуга* трактуется как графическое изображением обобщенной sc-дуги (ориентированной пары sc-элементов, семантика которой дополнительно уточняется явно средствами SC-кода);

- *основная sc.g-дуга* трактуется как графическое изображение sc-дуги основного вида (т. е. обозначения константной постоянно-положительной пары принадлежности);

- *sc.g-рамка 1-го вида* трактуется как ограничитель внешних информационных конструкций, визуально изображаемых в этой рамке, и является обозначением класса внешних информационных конструкций аналогичного вида;

- *sc.g-рамка 2-го вида* трактуется как ограничитель внешней информационной конструкции, визуально изображаемой в этой рамке, и является изображением sc-узла, обозначающего эту внешнюю информационную конструкцию.

Для широкого практического использования конструкций SCg-ядра не удобны, так как слишком громоздки и ненаглядны. Для решения этой проблемы предлагается использовать расширение SCg-ядра—SCg-код.

Наиболее часто используемые элементы пользовательского интерфейса визуализируются с помощью SCg-кода согласно семантической классификации sc-элементов [7].

Унификацию информационных объектов и объектов управления пользовательским интерфейсом можно рассматривать в двух аспектах — унификация визуального представления пользовательского интерфейса и унификация структур описания пользовательского интерфейса. Рассмотрим элементы управления пользовательского интерфейса, учитывая эти два аспекта.

3 ЭЛЕМЕНТЫ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Все элементы управления традиционного пользовательского интерфейса с точки зрения их назначения можно разбить на две большие группы [8]:

- управляющие окна — окна, используемые только для инициирования команд пользователя;
- обычные окна, используемые для просмотра различных информационных конструкций и инициирования команд пользователя.

Рассмотрим группу управляющих окон. К этой группе относят окна, которые позволяют пользователю, используя характерные для каждой группы окон алгоритмы реакций на действия пользователя, осуществлять управляющее воздействие на систему или другими словами передавать системе команды пользователя.

Традиционно сложилось, что алгоритмы реакции на действия пользователя у таких окон жестко реализованы и предусматривают модификацию только в рамках некоторого, предусмотренного разработчиками пользовательского интерфейса, шаблона. Как правило, такие шаблоны позволяют лишь расширить диапазон возможных реакций управляющего окна на воздействие пользователя на каждом шаге алгоритма, не меняя общей логики работы с управляющим окном. Среди управляющих окон можно выделить следующие классы окон:

- *окна-меню*, позволяют отобразить перечень классов команд и команды, имеющихся в распоряжении пользователя и служат для быстрого инициирования команд;

- *окна-кнопки*, позволяют отобразить пользователю произвольную команду и используются для быстрого инициирования команды;

- *окна-переключатели*, позволяют отобразить пользователю произвольную команду и используются для выбора единственного варианта из множества взаимоисключающих альтернатив;

- *окна-флаги*, позволяют отобразить пользователю произвольную команду и используются для выбора при наличии независимых вариантов выбора.

Обычные окна не обладают жестким алгоритмом реакций на действия пользователя. Реализации таких алгоритмов предоставляется разработчику пользовательского интерфейса.

По структуре оформления информации можно выделить следующие типы обычных окон:

- *окна-таблицы*, позволяющие отображать информацию в них в виде списков, имеющих неограниченное количество информационных полей (колонок);

- *окна-списки*, частный случай окон-таблиц (окна-таблицы с одним информационным полем), позволяющие

отображать информацию в них в виде обычных или выпадающих списков;

- окна с иерархической информацией, позволяющие отображать информацию в виде дерева с четкой фиксацией уровней;

- простые окна, окна позволяющие отображать информацию в текстовом либо графическом виде.

Для удобства пользователей в традиционном пользовательском интерфейсе используются различные группирующие окна. Примером таких окон может служить панель инструментов (обычное окно с размещенными на ней управляющими окнами) и группирующее окно (позволяет группировать).

В последние годы для достижения наилучшего когнитивного эффекта, всё больше и больше разработчиков пользовательских интерфейсов отходят от общепринятых норм. Эта тенденция привела к тому, что одинаково визуализированные пользовательские интерфейсы, нацеленные на решение одного и того же круга интерфейсных задач, позволяют решать эти задачи разными способами. Что, в свою очередь, вызывает нежелание у пользователя работать с подобными пользовательскими интерфейсами.

Таким образом, мы видим, что для пользовательских интерфейсов основным элементом управления является окно. Типология окон современного пользовательского интерфейса всегда определяется двумя свойствами -- характером, отображаемой в них информации, и реакцией окна на действия, производимые пользователем над окном.

Рассматривая подходы к унификации традиционных пользовательских интерфейсов, отметим, что существует ряд стандартов, определяющих визуальное представление пользовательского интерфейса, однако унифицированного подхода к описанию структур пользовательского интерфейса в памяти компьютера до сих пор не существует [8].

4 УНИФИКАЦИЯ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКИМ ИНТЕРФЕЙСОМ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В соответствии со сказанным выше, для пользовательского интерфейса sc-системы выделим два класса элементов пользовательского интерфейса -- окна sc-системы (или sc-окна) и ключевые узлы пользовательского интерфейса.

Окна sc-системы бывают разных видов, могут иметь разные типы и характеристики. Каждое sc-окно трактуется как sc-узел, отображенный на экране и имеющий содержимое. Отображение sc-узлов на экране осуществляется при помощи SCg-кода.

В зависимости от типа содержимого sc-узла различают текстовые, графические, речевые и другие sc-окна.

Окно пользовательского интерфейса sc-системы, содержимым которого является sc.g-конструкция, называется sc.g-окном пользовательского интерфейса sc-системы. Соответственно sc-окна, содержимым, которых является текст (строка или цепочка символов) какого-либо линейного языка (например, русского или английского),

называется текстовым sc-окном пользовательского интерфейса sc-системы. По аналогии вводятся другие типы sc-окон.

Окна пользовательского интерфейса sc-системы в зависимости от типа отображаемой на них информации делятся на два вида -- простые и гибридные.

К простым sc-окнам относятся окна, содержимое которых носит однородный характер, например, текст, рисунок, речь.

Гибридные sc-окна позволяют представлять неоднородную информацию. В частности гибридным может являться любое sc.g-окно, если оно содержит другие sc-окна.

В начале работы пользователя с sc-системой формируется sc-окно, которое называется главным (резидентным) sc-окном. Главное sc-окно принадлежит к классу sc.g-окон. Оно содержит всю вспомогательную информацию, необходимую пользователю для работы с sc-системой.

На множестве окон выделяют следующие отношения: быть дочерним окном, быть родительским окном, быть активным окном, быть закрытым окном (в этом случае окно отображается как sc.g-узел) и другие подобные отношения.

В базовые средства визуализации пользовательского интерфейса sc-систем включают следующие типы sc-окон: sc.g-окно, текстовое окно, окна для отображения линейной формы SC-кода и окна для отображения числовой информации.

В отличие от окон традиционного интерфейса, sc-окна не требуют включения в базовые средства пользовательского интерфейса специализированных типов окон (окон-списков, окон-таблиц и других подобных окон). Это обусловлено тем, что основной формой представления информации для sc.g-окон является семантическая сеть.

Большинство типов окон традиционного интерфейса было разработано с целью структурирования специализированной информации, отображаемой данным типом окон. Структуризация этой интерфейсной информации была необходима как пользователю, так и разработчику интерфейса. Однако структура интерфейсной информации, созданная разработчиком, редко совпадает со структурой интерфейсной информации, которую предполагает увидеть пользователь.

Одним из свойств семантических сетей является возможность наглядного показа концептуальной основы предметной области. Это свойство и позволяет отказаться от используемых в традиционном интерфейсе специализированных классов окон.

Другой тип элементов пользовательского интерфейса sc-систем используется для отображения и инициирования разнообразных интерфейсных команд.

Ключевые узлы пользовательского интерфейса sc-систем, отображаемые в главном sc-окне сразу после момента его создания, называют резидентными ключевыми узлами пользовательского интерфейса sc-систем.

Кроме главного окна, изображения знаков команд и классов команд могут быть доступны из других sc-окон. Графическое изображение знака команды или класса ко-

манды может меняться в зависимости от того, в sc-окне какого типа оно отображается.

Ключевые узлы пользовательского интерфейса могут являться знаками команд и знаками классов команд. Для того чтобы пользователь смог начать работу с командами определенного класса команд, ему необходимо инициализировать команду декомпозиции, аргументом для которой является знак некоторого класса команд. Команда декомпозиции входит в состав резидентных ключевых узлов пользовательского интерфейса.

Среди команд пользовательского интерфейса sc-систем выделяют особый класс команд – команды режимы. Команда режим - это команда, выполнение которой формирует sc-конструкцию, являющуюся описанием условий выполнения для некоторой другой группы команд. В этой группе команд могут находиться другие команды-режимы.

Для того чтобы сформулировать интерфейсную задачу пользователю необходимо создать sc.g-конструкцию, описывающую интерфейсную команду. Данная sc.g-конструкция должна содержать ключевой узел, указывающий тип команды, и узлы, которые должны являться аргументами команды.

Для решения интерфейсных задач в процессе работы с системой пользователь может сформулировать подряд несколько интерфейсных команд. Выполнение этих команд можно инициализировать двумя способами.

При первом способе, для решения интерфейсной задачи необходимо последовательно протранслировать формулировки команд в SC-код, то есть погрузить в базу знаний пользовательского интерфейса.

При втором способе, используя SCg-код необходимо описать протокол решения интерфейсной задачи. Для описания протокола необходимо описать формулировки команд и затем описать порядок выполнения этих команд пользовательским интерфейсом интеллектуальной системы. После описания протокола решения интерфейсной задачи его необходимо протранслировать в SC-код.

Минимальные средства общения интеллектуальной системы с пользователем при таком интерфейсе включают средства редактирования и отображения конструкций SCg-кода и средства трансляции конструкций SCg-кода в SC-код и обратно.

В отличие от традиционного пользовательского интерфейса, при использовании предлагаемого подхода, разработчик имеет возможность изменить логику работы с любым из элементов пользовательского интерфейса. Это свойство является результатом использования для описания пользовательского интерфейса высокоуровневого средства для представления интерфейсной информации – SC-кода.

Общение интеллектуальной системы и пользователя, с использованием такого интерфейса, происходит посредством пересылки фрагментов информационных конструкций записанных в SC-коде.

5 ЗАКЛЮЧЕНИЕ

В предложенном подходе для того, чтобы унифицировать визуальное представление пользовательского интерфейса, предлагается в качестве базового способа визуализации использовать SCg-код. SCg-код является открытым для расширения способом визуализации пользовательского интерфейса интеллектуальных систем.

Для того чтобы обеспечить унификацию структур представления пользовательского интерфейса интеллектуальных систем, предлагается преобразовать данные структуры в базу знаний, закодированную с помощью SC-кода.

Использование данного подхода при разработке пользовательского интерфейса интеллектуальных систем дает следующие преимущества:

- унификацию способа визуализации пользовательского интерфейса интеллектуальных систем;
- унификацию структур представления пользовательского интерфейса интеллектуальных систем;
- возможность в перспективе разработки интеллектуальных справочных систем, в БЗ которых интегрирована БЗ пользовательского интерфейса.

ЛИТЕРАТУРА

- [1] Раскин Д. Интерфейс: новые направления в развитии компьютерных систем: пер. с англ. /Д. Раскин.– СПб: Символ-Плюс, 2004.–272 с.
- [2] Грибова, В.В. Концепция разработки пользовательского интерфейса на основе онтологий. Часть 1. Инструментарий для разработки пользовательского интерфейса. Основная идея подхода./ В.В. Грибова, А.С.Клещев//Владивосток:ИАПУ ДВО РАН,2003,24с.
- [3] Джарратано, Д. Экспертные системы: принципы разработки и программирования: перс англ. / Д. Джарратано, Г. Райли.– М.: Диалектика-Вильямс, 2007.– 1152 стр.
- [4] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/В. Н. Касьянов, В. А. Евстигнеев//ВНУ–Санкт-Петербург, 2003.–1104с.
- [5] Гаврилова, Т.А. Визуальные методы работы со знаниями: попытка обзора/Т. А. Гаврилова, Н.А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, с. 15-21
- [6] Янкелевич, А. А. Применение графовых моделей для спецификации пользовательского интерфейса программных систем/ А. А. Янкелевич // Вестник молодых ученых 7'2001 серия: технические науки 2'2001. с.104–111
- [7] Представление и обработка знаний в графодинамических ассоциативных машинах / В.В. Голенков [и др.]; под ред. В.В. Голенкова.–Минск : БГУИР, 2001. – 412 с.
- [8] Гультяев, А.К. Проектирование и дизайн пользовательского интерфейса/ А. К. Гультяев, В.А. Машин.– 2-е изд. СПб.: КОРОНА принт, 2004.–352 с.