

Ю.И.Величко

МЕТОД АДАПТАЦИИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Введение. Пользовательский интерфейс (ПИ) играет важнейшую роль в функционировании человеко-машинной системы, поскольку от него зависит продуктивность взаимодействия оператора и информационного комплекса. За годы развития компьютерной техники были выработаны десятки эвристических правил нацеленных на создание «прозрачного» интерфейса, работа которого была бы интуитивно понятна любому пользователю. Но, несмотря на успехи в проектировании дружественных ПИ вопрос улучшения взаимодействия остается открытым, из-за того, что дизайнерские решения по проектированию интерфейсов ориентированы в большей степени на среднестатистического пользователя, а не на индивидуальные потребности каждого пользователя. Разумеется, что невозможно разработать интерфейс, который подошел бы любому человеку, так же как и не возможно разработать для каждого пользователя персональный интерфейс.

Частным решением такой проблемы является предоставление пользователю инструментов для модификации внешнего вида интерфейса согласно личным требованиям, а также отслеживание деятельности пользователя, и автоматической коррекции структуры интерфейса в зависимости от результата анализа этих действий. Это позволяет делать интерфейс более удобным в работе, но применение данных методов сводится к ограниченному кругу задач, и в каждом программном продукте реализовывается заново.

Общим решением проблемы создания «персонального» интерфейса может послужить внедрение в интерфейс аппарата адаптации. На сегодняшний день существуют системы и разработки в области реализации адаптивных интерфейсов. Но их реализация строго ограничена областью функционирования программы, для которой они разработаны и средствами реализации самого интерфейса. Поэтому адаптивные интерфейсы до сих пор не нашли широкого применения. Исправить ситуацию может универсальная система адаптации интерфейса (САИ), не зависящая от области применения и реализации интерфейса, способная проводить адаптацию любого интерфейса на абстрактном уровне, с последующей ее реализацией на конкретном интерфейсе.

Цель работы. Дать описание модели адаптации интерфейса не зависящей от предметной области и реализации интерфейса.

Модель пользователя. Ключевым компонентом, определяющим ход адаптации, является внутреннее представление системы о текущем пользователе, то есть модель пользователя (МП) [1]. В САИ было решено использовать стереотипную МП построенную по архитектуре ключ-значение, в которой «ключом» является название группы пользователей (ГП), а «значение» показывает степень принадлежности текущей модели к данной группе. Степень принадлежности определяется по характеристикам пользователя (ХП) полученным в процессе взаимодействия. Примерами характеристик пользователя может служить: пол, возраст, уровень интеллекта, уровень профессиональной подготовки и т.д.

Для описания модели пользователя основанной на нечетких показателях принадлежности к той или иной ГП, примем за множество возможных значений ХП, где - количество возможных значений ХП (все значения), а так же группы пользователей, о X пределяются экспертным путем на этапе реализации САИ). Тогда каждой ГП может быть поставлено в соответствие нечеткое множество пользователей A , такое, что для любого x выполняется условие[2]:

$$x \in X, \mu_A(x) \in [0,1] \tag{1}$$

В этом случае значение $\mu_A(x)$ будет показывать степень принадлежности МП к A -ой группе пользователей. Тогда модель пользователя можно представить в виде показанном на рис. 1.

	A	A	...	A
МП	0,75	0,3	...	0,0

Рис. 1. Модель пользователя при стереотипном подходе с нечеткой функцией принадлежности

Такая модель более точно отражает характеристики пользователя, за счет чего позволит системе адаптации производить изменение интерфейса основываясь наиболее приближенной к реальному человеку модели.

Модель адаптации. Для того что бы САИ была универсальной и зависела от реализации интерфейса, ее структура должна быть разделена на две составляющие: контроллер, задача которого состоит в регистрации событий интерфейса и его изменении, и ядро, которое реализует алгоритм адаптации. На верхнем уровне абстракции, работа системы адаптации состоит из четырех этапов: регистрация события, изменение МП, подготовка сценария адаптации, генерирование команд по изменению интерфейса (рис.2):

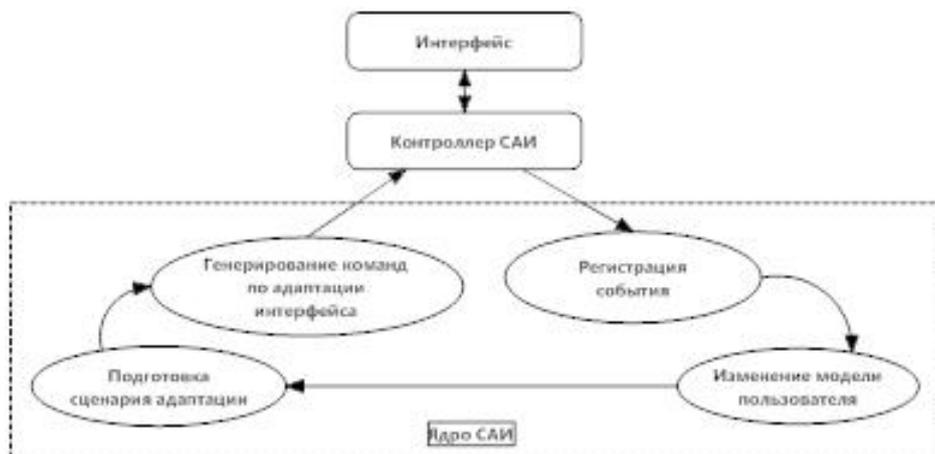


Рис.2. Обобщенная модель функционирования САИ

Разберем более подробно, как протекает каждый из этапов адаптации:

Событие – представляет собой возникновение некоего возмущающего фактора, который должен привести к переходу системы из одного состояния в другое, в соответствии с типом и параметрами события. Можно выделить два основных класса событий – внешние, и внутренние. Внешние события возникают в следствие определенных действий пользователей, в результате которых состояние интерфейса должно измениться. К таким событиям относятся, например, прохождение некоего теста определяющего текущие ХП, ручное изменение состояния интерфейса, или любое другое действие предусмотренное системой адаптации, которое требует изменения интерфейса. Внутреннее событие генерируется самой системой адаптации при возникновении соответствующих условий. Такие события зависят от мониторинга действий пользователя и основываются на внутренних алгоритмах выявления шаблонов поведения (регулярно повторяющихся последовательностей действий пользователя), что должно приводить к адаптации интерфейса. В результате выявления такого шаблона система адаптации самостоятельно генерирует событие, которое должно привести к изменениям в интерфейсе. Внешние события, как правило, не требуют оперативной ответной реакции САИ, и соответственно могут накапливаться на клиентской стороне, и передаваться пакетами в определенные моменты времени. Такой подход поможет уменьшить нагрузку на системы передачи информации, что особенно актуально в свете возможности использования САИ, на мобильных устройствах на которых качество и стоимость связи все еще оставляет желать лучшего. Этот алгоритм совершенно не противоречит логике адаптации, поскольку адаптация не является мгновенным процессом и в большинстве случаев требует времени для накопления сведений о пользователе.

Изменение МП – представляет собой пересчет значений записанных в МП, относящихся к одной или нескольким ХП которые были изменены. Поскольку предложенная МП не хранит текущие значения ХП, а только отражает принадлежность пользователя к той или иной ГП, система адаптации не может знать о том была ли на самом деле изменена ХП, или нет. С одной стороны это можно воспринимать как недостаток, поскольку возможны ситуации, когда процесс адаптации будет происходить без изменения интерфейса, или точнее в результате адаптации интерфейс меняться не будет. Однако предполагается, что подобные ситуации не должны случаться, поскольку сама суть возникновения событий, инициирующих модификацию МП, предполагает, что текущая МП не отражает пользователя в достаточной мере, и должна быть скорректирована. Соответственно проектировать систему необходимо так, что бы условия возникновения событий отражали тот факт, что текущая модель пользователя, и соответственно интерфейс который зависит от МП, не являются оптимальными. Тогда каждое событие будет свидетельствовать об необходимости корректировки МП. При этом нельзя забывать, что интерпретировать пожелания

пользователя и его действия необходимо с наивысшим приоритетом. Тогда те параметры интерфейса, которые пользователь изменил самостоятельно, не должны меняться под действием САИ без одобрения пользователя.

Подготовка сценария адаптации – этап, который заключается в извлечении из базы знаний САИ сценария адаптации соответствующего полученному событию. Сценарий адаптации представляет собой список действий по изменению интерфейса. Формально, действие можно представить в следующем виде:

«Изменить размер шрифта всех кнопок на X »

где – X величина, которая рассчитывается САИ на основе текущей МП. Сценарий адаптации может состоять из нескольких подобных шагов для разных компонентов интерфейса.

Генерирование команд для изменения интерфейса – окончательный этап в работе ядра системы адаптации. Он состоит в определении новых значений свойств, или нового порядка расположения элементов интерфейса. Для этого система адаптации рассчитывает необходимые для текущего сценария значения величины X . Правила для расчета имеют вид $X = f(d_0, d_1, \dots, d_n)$, где d_i - параметр интерфейса, который может принимать либо значимое (*const*) либо неопределенное (*NULL*) значение; функция $f(d_i, \dots)$ - определяет значение X основываясь на значениях входных переменных и внутреннего алгоритма, зависящего от характера вычисляемого параметра.

Параметры d_i вычисляются на основании текущей МП, с помощью механизма нечеткой базы знаний параметров компонентов интерфейса (БЗПКИ), составленной экспертами, под которой будем понимать совокупность нечетких правил «Если-То», определяющих взаимосвязь между параметрами модели пользователя и значениями свойств компонентов интерфейса. Тогда обобщенный формат нечетких правил будет иметь вид:

IF посылка правила, *THEN* заключение правила.

Посылка правила или антецедент представляет собой утверждение типа « $x \in A_i$ » и означает, что модель пользователя относится к определенной группе пользователей. В данной посылке A_i - это терм определяющий имя группы пользователей заданной нечетким множеством на множестве возможных значений определенной ХП, а x - условно представляет модель пользователя. Словесно посылку можно описать так – «текущая модель входит в группу». Заключение, или следствие правила представляет собой утверждение типа $y = d$ в котором значение выходной переменной d задается четкой константой – значением определенного параметра компонента интерфейса. При таком подходе БЗПКИ может быть представлена в следующем виде:

$$\left(\bigcap x = A_i^j \right) \rightarrow y = d_k, j = [0, n], k = \overline{1, m} \tag{2}$$

Такую базу знаний легко представить в табличном виде:

x	x	x	x		x	y
a_1	a_3					d_1
a_7	a_{13}	a_{18}	a_{33}	a_8		d_1
...
a_5	a_1	a_7	a_9			d_k

Данная БЗ может отражать алгоритм определения свойств компонентов интерфейса любой реализации, так как оперирует не конкретными элементами, а абстрактными. Недостатком такой организации данных может быть наличие большого количества пустых ячеек при добавлении правил зависящих от большого количества значений МП.

Не исключено, что при проектировании БЗПКИ значение одного параметра будет зависеть от нескольких правил. В таком случае база знаний примет вид:

$$\bigcup_{p=1}^{k_j} \left(\bigcap_{j=1}^n (x_i = A_i^{jp}) \right) \rightarrow y = d_j \tag{3}$$

где A_i^{jp} - группа пользователей, к которой определяется принадлежность текущей модели пользователя x_i в строчке с номером $jp, p = \overline{1, k_j}$; k_j - количество строчек-конъюнкций, в которых выход y

оценивается нечетким термом d_j , $j=1,m$; m - количество правил, используемых для определения параметра компонента интерфейса.

Заключение. При совместном использовании модели пользователя основанной на показателях нечеткой принадлежности к множеству групп пользователей, и предложенной в работе модели адаптации интерфейса, можно создать САИ не зависящую от реализации интерфейса, предметной области использования системы и потенциальных пользователей. Это достижимо за счет реализации алгоритма адаптации на абстрактном, не привязанном к реализации интерфейса, уровне.

ЛИТЕРАТУРА

1. Ходаков В. Е., Ходаков Д. В. Адаптивный пользовательский интерфейс: проблемы построения // Автоматика. Автоматизация. Электротехнические комплексы и системы №2 (11). 2003 – С. 45-57
2. Яхьяева Г.Э. Нечеткие множества и нейронные сети. Серия: Основы информационных технологий. – БИНОМ, 2008. – 320 с.

ВЕЛИЧКО Юрий Игоревич – ассистент кафедры информационных технологий ХНТУ.

Научные интересы: адаптивные пользовательские интерфейсы.