Нечеткий процессно-ориентированный подход к недетерминированному проектированию интеллектуальных мультимодальных интерфейсов

11, ноябрь 2012

DOI: 10.7463/1112.0495687

Алфимцев А. Н.

УДК 004.55

Россия, МГТУ им. Н.Э. Баумана

alfim@bmstu.ru

1. Введение

Актуальность. Проектирование интеллектуальных мультимодальных интерфейсов (ИМИ) может проводиться с учетом различных критериев качества созданного проекта: эргономический, вычислительный, экономический, интеллектуальный, мультимедийный, мультимодальный и др. [1, 2, 3]. При этом принятие решения проектировщиком о выборе окончательного варианта проекта производится не только в условиях многокритериальности, но и в условиях неопределенности, причинами которой являются:

I. Сложность и трудоемкость определения точных значений ряда показателей, вызванная небольшим объемом выделяемых ресурсов на предварительное исследование и анализ целей заказчика. В результате чего производные требования и критерии качества (не заявленные прямо заказчиком, а подразумеваемые или вытекающие из задачи) могут в итоге составить 95% всех требований к разрабатываемому программному обеспечению [8, 9];

II. Необходимость прогнозирования изменений условий решаемой задачи (количество модальностей, типы графического интерфейса, правила бихевиоризации элементов интерфейса и т.д.) из-за значительного временного разрыва, по меркам отрасли разработки программного обеспечения, от начала проектирования до принятия решения по внедрению интерфейса. При этом ошибка в прогнозе, и в результате будущая модификация программного кода более чем на 20-25 % приводит к переписыванию кода заново [7];

III. Уникальность, присущая в некоторой степени всем проектируемым пользовательским интерфейсам, являющимся сегодня сложными человеко-машинными системами, особенно для тех систем, которые функционируют в области интеллектуального взаимодействия. При этом программное обеспечение, которое было тщательно протестировано с покрытием кода до 90 %, может содержать дефекты, 35 % из которых вызвано логическими ошибками, 40 % - уникальными комбинациями этих ошибок [10].

Поскольку задача проектирования решается в условиях неопределенности, следовательно, возрастает вероятность того, что созданный проект будет не удовлетворять выбранным критериям, что приведет к новым итерациям проектирования, рефакторингу, провалу или не завершению проекта в отведенный период. По данным компании *Standish Group* только 16 % проектов по разработке программного обеспечения завершаются в срок [11]. Поэтому, актуальными с теоретической и практической точек зрения, является задачи формализации процессов проектирования и определения частных критериев эффективности для разрабатываемого ИМИ. Данные задачи могут быть решены с помощью методологии детерминированного проектирования интеллектуальных мультимодальных интерфейсов [3]. Однако выбирая один вариант и сужая множество альтернатив проекта непосредственно в процессе разработки, проектировщик отбрасывает другие варианты, и не дает возможности в будущем осуществить формальную многокритериальную проверку по всем возможным вариантам проектируемого интерфейса.

Решением данной проблемы является предоставление проектировщику способа создания проектного решения, одновременно состоящего из нескольких допускаемых вариантов (удовлетворяющих техническим и эксплуатационным требованиям и заданным ограничениям) на каждом этапе проектирования. Для этого может быть использован формальный язык процессов в качестве языка моделирования проекта ИМИ и нечеткая логика, допускающая явное описание степени выполнимости того или иного процесса, описывающего функционирование ИМИ.

Обзор предшествующих работ и место статьи.

Интеллектуальный мультимодальный интерфейс предоставляет пользователям возможность интеллектуального взаимодействия адекватно их модальностям. В области искусственного интеллекта разработано ряд методологий, которые могут быть использованы для разработки именно интеллектуального взаимодействия [16]: *Gaia*, *Mase*, *Graph Theory Methodology*, *UML*, *IDEF8*, *Design Patterns*, *Design Components*, *Modeling*

Database Information Systems. Однако главными недостатками этих подходов являются низкий уровень формализации, противоречивость описания, сложность или высокая степень абстрагированности.

Поэтому для автоматизированного проектирования ИМИ в основу методологий закладываются методы, которые явно используют формальные модели проектирования [1, 2, 17]. Автоматизация проектирования процессов функционирования человеко-машинных систем в работе [1] основывается на вероятностных и нечетких показателях. В работе [2] для автоматизации проектирования, реализации и сопровождения пользовательского интерфейса применяется онтологический подход. Работа [17] в целом посвящена методологии разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники. Кроме того, помимо работ прямо направленных на формализацию и автоматизацию проектирования процессов программного обеспечения, в настоящее время активно развиваются методы формализации проектирования бизнес-процессов в различных областях применения [15], которые также могут быть использованы для создания формализованной методологии проектирования ИМИ.

Общими недостатками всех этих подходов является ориентация, как правило, на эрагатическую составляющую интерфейса или на одну формальную модель, что существенно ограничивает возможности проектирования. Для устранения этого недостатка в основу методологии создания ИМИ необходимо положить такую совокупность взаимосвязанных формальных моделей и операций над ними, которые позволяли бы решать весь комплекс задач проектирования [16]. В частности, эти модели должны поддерживать иерархическое описание методологии проектирования ИМИ, композицию одних процессов из других, обоснованный переход от одних моделей к другим, проверку синтаксической корректности, функциональной правильности, выполнение эквивалентных и оптимизирующих преобразований, адекватное представление информации в зависимости от этапа и уровня проектирования [3].

В настоящей работе развивается данный процессно-ориентированный подход к созданию формальных моделей в зависимости от решаемой задачи и рассматриваются принципы его использования для формализации недетерминированного проектирования ИМИ.

Цель статьи и постановка задач. Целью данной работы является разработка нечеткого процессно-ориентированного подхода для недетерминированного проектирования интеллектуального мультимодального интерфейса. Проект ИМИ задается

в формальной системе: $\Phi C_1 = \{Y_1, P_1, M\}$, где Y_1 – язык процессов, использующийся для описания проекта ИМИ, P_1 – правила на языке процессов, представляющие собой непосредственно проект ИМИ, M – абстрактная машина вывода.

Для реализации этой цели были поставлены и решены следующие задачи:

1. Определить понятия недетерминированного и нечеткого выполнения процессов проекта ИМИ заданного в формальной системе ΦC_1 .
2. Разработать метод выбора нечетко выполнимых процессов в проекте ИМИ.
3. Разработать визуальный язык представления процессов, который позволит проектировщику создавать проект ИМИ, используя инструменты графического пользовательского интерфейса.
- 4.

Научная новизна. В статье предложен новый нечеткий процессно-ориентированный подход к недетерминированному проектированию интеллектуального мультимодального интерфейса, включающий понятия недетерминированного и нечеткого выполнения процессов проекта ИМИ и метод выбора нечетко выполнимых процессов в проекте ИМИ.

Структура статьи. Далее структура статьи следующая. В разделе 2 вводятся основные элементы теории процессов. В разделе 3 дается определение недетерминированного и нечетко выполнимого процесса. В разделе 4 описывается метод выбора варианта проекта ИМИ. В разделе 5 рассматривается визуальный язык представления процессов.

2. Определение понятия процесса в проекте ИМИ

С каждым процессом связан его *алфавит восприятий и реакций* $Act = \{a_1, a_2, \dots, a_m\}$. Каждый символ a этого алфавита именуется некоторый объект, получаемый (воспринимаемый) процессом из внешней среды (*восприятие процесса*), выдаваемый процессом во внешнюю среду (*внешняя реакция процесса*) или объект, используемый процессом для внутренних нужд (*внутренняя реакция процесса*). Процессы действуют, воспринимая, порождая для внутреннего употребления или выдавая наружу объекты с соответствующими именами. Для того, чтобы различать типы действий будем использовать следующие обозначения: $?a$ для восприятий, $!a$ – для внешних реакций, τa или просто τ – для внутренних реакций.

Нитью a^* будем называть кортеж (конечный или бесконечный) действий $\langle a_0 a_1 a_2 \dots a_{m-2} a_m \rangle$, т.е. $a^* = \langle a_0 a_1 a_2 \dots a_{m-2} a_m \rangle$. Выполнением нити называется последовательность выполнения действий в порядке их записи в нити слева направо, т. е. осуществление в порядке слева направо по порядку восприятия или реакции. Символом e обозначим пустое действие. Нить, состоящая из единственного пустого действия, называется *пустой нитью*.

Процессом P называется множество нитей $Th(P)$, которые он может выполнять. Поведением процесса P называются правила выполнения нитей, принадлежащих множеству $Th(P)$. Эти правила запишем на языке регулярных выражений [4]. Если нить $a^* = \langle a_0 a_1 a_2 \dots a_{m-2} a_m \rangle$ конечна, то ее описание определяется как $P' \triangleq a_0 . a_1 . a_2 \dots a_{m-2} . a_m . 0$, где знак $\triangleq$ обозначает равенство по определению, все действия разделяются точками, P' – имя процесса такого, что множеству $Th(P')$ принадлежит нить a^* и все ее начала, а 0 – пустой процесс, т.е. $Th(0) = \emptyset$. Наличие пустого процесса после действия a_m означает конец нити a^* (после действия a_m не следует никаких действий). Если процесс слишком длинный, например, $P' \triangleq a_1^a . a_2^a \dots a_{m_a}^a . a_1^b . a_2^b \dots a_{m_b}^b \dots a_1^t . a_2^t \dots a_{m_t}^t . 0$, то он может быть представлен как кортеж процессов: $P' \triangleq P^a P^b \dots P^t$, где $P^a \triangleq a_1^a . a_2^a \dots a_{m_a}^a . 0$, $P^b \triangleq a_1^b . a_2^b \dots a_{m_b}^b . 0$, ..., $P^t \triangleq a_1^t . a_2^t \dots a_{m_t}^t . 0$.

Когда процессы $P^a = a_0 . a_1 . a_2 \dots a_{m-2} . a_m . a_1^a . a_2^a \dots a_{m_a}^a . 0$, $P^b = a_0 . a_1 . a_2 \dots a_{m-2} . a_m . a_1^b . a_2^b \dots a_{m_b}^b . 0$, ..., $P^t = a_0 . a_1 . a_2 \dots a_{m-2} . a_m . a_1^t . a_2^t \dots a_{m_t}^t . 0$ имеют общее начало $P' \triangleq a_0 . a_1 . a_2 \dots a_{m-2} . a_m$, то они могут быть представлены процессным выражением $P' \triangleq P' (P^a | P^b \dots | P^t)$, $P^a \triangleq a_1^a . a_2^a \dots a_{m_a}^a . 0$, $P^b \triangleq a_1^b . a_2^b \dots a_{m_b}^b . 0$, ..., $P^t \triangleq a_1^t . a_2^t \dots a_{m_t}^t . 0$. Если процесс P'' заиклиивается, то для его описания будем использовать регулярное выражение $\{P''\}$. Если заиклиивание происходит не с начала, а после некоторой нити a^* , то для описания используется регулярное выражение $P'' = a^* . \{P''\}$. Если $P'' \triangleq (P^a | P^b \dots | P^t)$, то $\{P''\} = (\{P^a\} | \{P^b\} \dots | \{P^t\})$. Если $P'' = P' (P^a | P^b \dots | P^t)$, то $\{P''\} = (\{P' P^a\} | \{P' P^b\} \dots | \{P' P^t\})$. $P''' \triangleq (P^a || P^b \dots || P^t)$ – процесс P''' , состоит из процессов $P^a, P^b, \dots, P^t$, выполняющихся параллельно.

Подчеркивая дуальность формализации проектирования ИМИ, отметим, что в ΦC_1 , основанной на теории процессов, проверяется процедурная семантика проекта ИМИ, тогда как для проверки декларативной семантики может быть задана метаформальная система: $\Phi C_2 = \{Y_2, P_2, M_2\}$, где Y_2 – язык модальной логики, использующийся для описания свойств ИМИ, P_2 – свойства ИМИ на языке модальной логики, M_2 – некоторая абстрактная машина вывода. Примером реализации метаформальной системы в качестве отдельного модуля архитектуры агента являются гибридные МАС [5]. Но, чтобы упростить структуру каждого процесса в процессно-ориентированной методологии проектирования ИМИ введем новый элемент: Π – предикат. Тогда в кортеже процессов вида: $P' \triangleq P^a \Pi P^b \dots P^t$, будет выполняться вначале процесс P^a , а процесс P^b будет выполнен, только после выполнения предиката Π и всех запущенных от него процессов.

3. Недетерминированное и нечеткое выполнение процессов

В данном случае основным принципом, на котором будет базироваться недетерминированное проектирование ИМИ, является принцип выбора из двух вариантов. То есть, после создания каждого нового процесса проектировщик имеет возможность задать до двух групп процессов, которые будут выполняться после этого процесса. Обоснованием именно двух вариантов, служат следующие факты. Во-первых, несмотря на то, что человеческий мозг способен одновременно удерживать в поле внимания 3-7 объектов, из теории психологии выбора известно, что человек всегда старается свести множество вариантов выбора к выбору из двух вариантов [13]. И, во-вторых, при использовании двух возможных вариантов дерево выбора является двоичным и верхняя оценка вычислительной сложности расхода памяти компьютера и выполнения операций поиска, вставки и удаления составит всего лишь $O(h)$, где h – глубина дерева.

Недетерминированным выполнением процессов ($\oplus$) будем называть возможность выполнения после некоторого процесса P^a процесса P^b или P^t : $P^a P^b \oplus P^t$. Использование скобок, например $P^a (P^b P^c \dots P^k \oplus P^s P^t \dots P^w)$, означает что после процесса P^a будут выполняться процессы $P^b P^c \dots P^k$ или $P^s P^t \dots P^w$. Для того чтобы проектировщик мог явно задать степень выполнимости того или иного процесса введем понятие нечеткого множества выполнимости процессов: $A_p = \{x, \mu_A(x) \mid x \in X\}$, где X – множество процессов, стоящих слева и справа от оператора $\oplus$, $\mu_A(x)$ – функция принадлежности, характеризующая степень принадлежности некоторого процесса P' нечеткому множеству A_p , принимающая свои значения в $[0, 1]$. Тогда нечетким выполнением процессов ($\oplus_z$)

будем называть возможность выполнения после некоторого процесса P^a процессов $P^b P^c \dots P^k$ со значением z или $P^s P^t \dots P^w$ со значением $(z-1)$: $P^a(P^b P^c \dots P^k \oplus_z P^s P^t \dots P^w)$, где оператор $\oplus_z$ параметризован числом $z = \mu_A(x)$. Например, запись $P^a P^b \oplus_{0.4} P^t$ будет обозначать, что со степенью выполнимости равной 0.4 после процесса P^a выполняется процесс P^b и со степенью 0.6 процесс P^t . В случае $(P^b \oplus_{0.7} P^t) \parallel P^k P^l \dots P^m$ со степенью 0.7 выполнятся $P^b \parallel P^k P^l \dots P^m$ и со степенью 0.3 выполнятся процессы $P^t \parallel P^k P^l \dots P^m$.

Однако проектировщику не всегда удобно оперировать с числовыми данными, поэтому определим понятие лингвистической переменной, которое позволит задавать степень выполнимости процесса словами или выражениями естественного языка. Базовым множеством для всех переменных будет являться множество действительных чисел R . Множество его нечетких подмножеств $F(R)$ будем называть множеством нечетких чисел. Так же, будем считать, что функция принадлежности $\mu_A: R \rightarrow [0,1]$ нечеткого числа задает ограничения на соответствие элементов R этому нечеткому числу.

Если некоторым элементам $F(R)$ ставятся в соответствие слова или выражения естественного языка, выражающие плохо определенные понятия, то эти элементы образуют множество лингвистических значений $L(R) \subset F(R)$. Предположим, что каждому элементу $F(R)$ соответствует слово или выражение естественного языка, т.е. $L(R) = F(R)$. Поэтому все множество нечетких чисел будем называть *лингвистическим*. Переменную, определенную на базовом множестве R будем называть базовой, а на лингвистическом - лингвистической. В соответствии с этим определением, четким значением базовой переменной является элемент R , аналогично для лингвистической – элемент $F(R)$. Нечеткое подмножество $F(R)$ значений лингвистической переменной называют также нечетким множеством второго уровня [12].

Нечеткие множества второго уровня могут интерпретироваться как нечеткие высказывания вида: " x есть l есть ξ ", где: l - лингвистическое значение; ξ - истинность этого значения. Таким образом, нечеткое значение $\tilde{l}$ лингвистической переменной l можно представить как: $\tilde{l} = \bigcup_{l \in F(R)} \xi(l)/l$, где $\xi: F(R) \rightarrow [0,1]$ - функция принадлежности второго уровня. Кроме того, над лингвистическими переменными могут быть определены лингвистические операции, главными из которых являются лингвистические

модификаторы (унарные операции) и связки (бинарные операции). Их важной особенностью является то, что они могут служить эффективным инструментом для конструирования выражений естественного языка. Но, естественный язык не является формальным, и здесь используется только для представления степеней выполнимости процессов. В качестве формального языка проекта ИМИ используется язык процессов, хотя сегодня в проектировании ИМИ, как это будет показано в разделе 5, необходимо иметь возможность использовать и некоторый простой визуальный язык.

4. Метод выбора варианта проекта ИМИ

Использование нечетко выполнимых процессов предоставляет возможность выбирать окончательный вариант проекта ИМИ в зависимости от вычисленных критериев. То есть, задача выбора проекта сводится к задаче оценки наилучшего проектного решения. Для этого может быть использован следующий метод выбора варианта проекта ИМИ:

Шаг 1. Создание недетерминированного проекта интеллектуального мультимодального интерфейса на основе нечеткого процессно-ориентированного подхода. Этот шаг также можно назвать проходом вперед, так как экспертом или на основе шаблонов проектирования задаются функции принадлежности, и тем самым количественно определяются возможные варианты проекта ИМИ.

Шаг 2. Выбор критериев. На данном шаге необходимо выбрать критерии, по которым будет оцениваться готовый проект ИМИ. Эти критерии могут быть связаны с проверкой будущей рентабельности создаваемого программного продукта, дружелюбности к пользователю или же проверки его интеллектуальных свойств, которые применительно к ИМИ включают свойства: обязательной реакции, обоснования, обучения, оптимизируемости, модифицируемости, коллективности, безопасности, мобильности, скрытого постоянства, рекомендательности.

Шаг 3. В цикле для каждого критерия выполняются последующие шаги 4-6.

Шаг 4. Проверка критерия. Как было отмечено в формальной системе ΦC_1 , может быть проверена процедурная семантика, а декларативная семантика проверяется с помощью ΦC_2 .

Шаг 5. Нормирование значения критерия. Если критерий имеет числовой показатель, то его необходимо фаззифицировать для шага 5. Если же критерии носят скорее качественную оценку, то может быть использована следующая формула по

нормированию для каждого критерия:
$$\text{Поправка} = \frac{\text{Значение критерия} \cdot \text{Вес критерия}}{\text{Количество критериев}}.$$

Например, для критериев в виде свойств ИМИ смысл формулы будет заключаться в увеличении значений степеней выполнимости процессов на 0.1 для тех вариантов решений проекта ИМИ, которые удовлетворяют данному свойству.

Шаг 6. Обратный проход. В результате обратного прохода, по аналогии с алгоритмом обратного распространения ошибки в нейронных сетях, к степеням выполнимости процессов прибавляется значение поправки, если критерий выполнен для данного варианта проекта ИМИ и его значение вычитается, если критерий не выполнен.

Шаг 7. В случае достижения заданных пороговых показателей для степеней выполнимости процессов проекта ИМИ – переход к шагу 8, в ином случае, переход к шагу 1.

Шаг 8. Переход к этапу реализации проекта ИМИ в методологии проектирования [3].

Верхняя оценка вычислительной сложности данного метода составит $O(kn^2)$, где k - количество проверяемых критериев и n - количество процессов проекта.

5. Визуальный язык представления процессов

Современная разработка программного обеспечения, и в частности человеко-машинных интерфейсов, на основе некоторого визуального языка проектирования является стандартом «де факто»: IBM Rational Rose, Oracle Process Architect, ИНТЕЛЛЕКТ-2 и др. Поэтому нечеткий процессно-ориентированный подход для недетерминированного проектирования ИМИ был реализован в программном продукте ПИМИ, зарегистрированном в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам как программа для ЭВМ №2011613303. Графический интерфейс ПИМИ представлен на рис. 1.

Рис. 1. Графический интерфейс ПИМИ

Процессно-ориентированное проектирование в нем производится с использованием *визуального языка представления процессов (ВЯПП)*. Формально визуальный язык представления процессов - есть семерка:

$$G = \langle \Sigma, \Psi, \Omega, \Phi, X, Y, Z \rangle,$$

где: Σ - скелетная схема; Ψ - таблица степеней выполнимости; Ω - таблица преобразования информации; Φ - таблица фактических выражений; X, Y, Z - множества соответственно входных, внутренних и выходных переменных ВЯПП G .

По аналогии с языком операторных схем параллельных алгоритмов с памятью (ОСПАП) [14], скелетная схема Σ представляет собой граф из визуальных блоков, соединенных друг с другом дугами. Блоки на скелетной схеме имеют только входы и выходы. Существуют пять видов блоков: Н - стартер, имеет только один выход; Р - разветвитель, имеет один вход и не менее двух выходов; О - останов, имеет один или более вход и ни одного выхода; Р - процесс, имеет не менее одного входа и не менее одного выхода; П - предикат, имеет не менее одного входа и один выход (см. таблицу 1).

Из перечисленных видов блоков скелетная схема Σ ВЯПП строится с учетом следующих ограничений: стартер всегда один; каждый вход любого блока соединен дугой только с входом другого блока и только выход предиката может соединяться с его же входом непосредственно; не может быть, кроме стартера, блоков без входа, и кроме останова, блоков без выхода. Некоторый произвольный блок из числа перечисленных будем обозначать S . Если выход какого-либо блока S_i соединен дугой с входом блока S_j , то на скелетной схеме имеется дуга (S_i, S_j) , соединяющая эти вход и выход. Если блок S_i имеет q выходов, соединенных со входами блоков $S_{j_1}, \dots, S_{j_q}$, то совокупность дуг $(S_i, S_{j_1}), \dots, (S_i, S_{j_q})$ называется фрагментом и может записываться как $(S_i, S_{j_1}, \dots, S_{j_q})$.

Таблица степеней выполнимости Ψ задает для каждого процесса и предиката ВЯПП соответствующие нечеткие значения или лингвистические переменные для связей этих блоков в скелетной схеме Σ , определяющих степень выполнимости того или иного процесса и предиката. Эти значения таблицы могут быть явно указаны на связях блоков ВЯПП.

Таблица преобразования информации Ω задает для каждого процесса и предиката ВЯПП множество его входных переменных $u_1, \dots, u_p$ из числа входных и внутренних переменных ВЯПП: $\{u_1, \dots, u_p\} \subseteq X \cup Y$, множество выходных переменных $w_1, \dots, w_r$ из числа внутренних и выходных переменных этой ВЯПП: $\{w_1, \dots, w_r\} \subseteq Y \cup Z$ и множество отображений $f_i, i = 1, \dots, k$, значений переменных $u_1, \dots, u_p$ в значения переменных $w_1, \dots, w_r$, каждое из которых записывается в общем случае как $(w_1, \dots, w_r) := f_i(u_1, \dots, u_p)$.

Множество всех преобразований $f_i, i = 1, \dots, k$, соответствующих одному блоку S , который может быть процессом или предикатом, будем записывать как $S \leftrightarrow f_1, \dots, f_k$. Если S является предикатом Π , входящим в фрагмент $(\Pi, S_1, \dots, S_s)$, то $f_i, i = 1, \dots, k$ соответствует дуге (Π, S_i) и имеет вид $w_i := f_i(u_1, \dots, u_p)$.

В таблице фактических выражений Φ задается конкретный вид всех преобразований описанных в Ω , в виде формул, связывающих входные и выходные переменные. Набор операций, содержащихся в формулах может варьироваться.

В общем случае блоки ВЯПП образуют иерархию частных блоков ВЯПП, на нижнем уровне которой находятся блоки ВЯПП содержащие только простые процессы и предикаты. Предполагается, что процессы, описанные на языке ВЯПП, выполняются во времени, которое дискретно, и состоит в переработке блоками ВЯПП одного множества объектных переменных в другое. В каждый момент времени (такт) может выполняться один или несколько блоков ВЯПП.

В таблице 1 представлено сопоставление базовых конструкций ВЯПП, который используется для проектирования ИМИ, элементов теории процессов, использующийся для моделирования проекта ИМИ, языка регулярных схем алгоритмов (РСА) [15], языка *BPEL* [6], который позиционируется как стандарт в компьютерном представлении процессов и элементов языка *Prolog*, который может использоваться в качестве языка логического программирования на котором задаются и машиной вывода проверяются правила проекта ИМИ.

Таблица 1.

Сопоставление способов описания проекта ИМИ

Функцио- нальное назначение	ВЯПП	BPEL	PCA	Язык процессов	Prolog
1	2	3	4	5	6
Стартер		<code><process name="..." targetNames pace="..." ></code>	$R=$	$P \triangleq \dots$	<i>Begin()</i>
Процесс		<code>... name=" 1 P "...</code>	P	P	P
Предикат		-	-	Π	<i>Pred()</i>
Разветви- тель		<code><if name=" " 1 x "> <condition> "1" </condition> > <invoke name=" 1 P " ... /> <else> < invoke name=" 2 P " ... /> </else> </if></code>	$_{x_1} (P_1 \wedge P_2)^{x_1}$	$\oplus_z$	<i>Or (Pa,Pb)</i>
Последова- тельное выполнение		<code><sequence... "> <invoke name=" 1 P " ... /> <invoke name=" 2 P " ... /> </sequence></code>	$P_1 \cdot P_2$	$P' \triangleq P^a P^b \dots P^t$	<i>Transition(Pa,Pb, ...,Pt)</i>
Параллель- ное выполнение		<code><flow ...> <invoke name=" 2 P " ... /> <invoke name=" 2 P " ... /> </flow></code>	$[P_1 \wedge P_2]$	$P''' \triangleq (P^a \parallel P^b \dots \parallel P^t)$	<i>Parallel(Pa,Pb,...,Pt)</i>

Восприятие процесса		<pre><process name=" 1 P "> <partnerLink name=" c"> partnerLink Type="участующий" myRole=" 2 P " /> </process></pre>	-	?a	In(P,a1,...,am)
Внешняя реакция процесса		<pre><process name=" 2 P "> <partnerLink name=" c"> partnerLink Type="выходящий" myRole=" 1 P " /> </process></pre>	-	!a	Out(P,a1,...,am)
Внутренняя реакция процесса		<pre><process ... /> </process></pre>	-	τa	Tau(a1,a2,...,am)
Останов		<pre></process></pre>	.	.	End()

6. Заключение

В статье описан нечеткий процессно-ориентированный подход к недетерминированному проектированию интеллектуальных мультимодальных интерфейсов с использованием визуального языка представления процессов. Главные достоинства и отличия предлагаемого подхода от известных аналогов следующие:

- возможность одновременной разработки нескольких вариантов проекта ИМИ и автоматического выбора варианта в зависимости от выполнимости критериев или свойств ИМИ;
- возможность вычисления нечеткой степени выполнимости процессов проекта ИМИ;
- возможность визуального представления проекта ИМИ, интуитивно понятного проектировщику и представления в виде системы уравнений, компактно описывающих формальную часть проекта и использующихся для решения следующих задач: оптимизации, определения функциональной правильности, проверки на синтаксическую и семантическую корректность, осуществления автоматической коррекции

спроектированного ИМИ в процессно-ориентированной нотации на основе проверки эквивалентности с любым эталонным процессом.

Список литературы:

1. Гриф М.Г. Автоматизация проектирования процессов функционирования человеко-машинных систем по вероятностным и нечетким показателям : дис. ... докт. техн. наук. Новосибирск, 2002. 270 с.
2. Грибова В.В. Автоматизация проектирования, реализации и сопровождения пользовательского интерфейса на основе онтологического подхода : дис. ... докт. техн. наук. Владивосток, 2007. 393 с.
3. Алфимцев А.Н., Девятков В.В. Принципы формализации методологии детерминированного проектирования интеллектуального мультимодального интерфейса // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2012. Спец. вып. № 4 Компьютерные системы и технологии. С. 160-176.
4. Kleene S.C. Representation of Events in Nerve Nets and Finite Automata // Automata Studies / C. E. Shannon, J. McCarthy, W. R. Ashby. Princeton University Press, 1956. Ch.1. P. 3–42.
5. Morris A., Ulieru M. FRIEND: A Human-Aware BDI Agent Architecture // Proc. of the International Conference «Systems, Man and Cybernetics». Anchorage: IEEE, 2011. P. 2413-2418.
6. BPEL Cookbook: Best Practices for SOA-based integration and composite applications development / J. Bolie, M. Cardella, S. Blanvalet, M. Juric, S. Carey, P. Chandran, Y. Coene, K. Geminiuc, M. Zirn, H. Gaur. Birmingham-Mumbai: Pact publishing, 2006. 185 p.
7. Гласс Р. Факты и заблуждения профессионального программирования : пер. с англ. / под ред. А. Галунова. СПб.: Символ-Плюс, 2007. 240 с. [Glass R.L. Facts and Fallacies of Software Engineering. New York: Addison-Wesley, 2004.].
8. Brooks Jr. F.P. The Design of Design: Essays from a Computer Scientist. New York: Addison-Wesley, 2010. 400 p.
9. Glass R.L. Building Quality Software. New Jersey: Prentice Hall, 1992. 369 p.
10. Макконнелл С. Совершенный код. Мастер-класс : пер. с англ. / под ред. А.И. Козлова. М. : Издательско-торговый дом «Русская Редакция», СПб.: Питер, 2005. 896 с. [McConnell S. Code Complete. Redmond: Microsoft Press, 2004.].

11. Sutherland J. Managing the iterative process. Режим доступа:
<http://www.standishgroup.com/> (дата обращения 09.10.2012).
12. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning // Information Sciences. 1975. № 8. P. 199-249.
13. Латыпов И. Психология выбора. Режим доступа:
<http://www.gptc.ru/index.php/publications/124-psychology-of-choice.html> (дата обращения 09.10.2012).
14. Девятков В.В. Технология автоматизированного проектирования систем программно-логического управления // Автоматизация проектирования: сб. науч. тр. М., 1990. С. 4 – 28.
15. Илюшко В., Калмыков А. Формализация бизнес-процессов на основе аппаратов РСА и π -исчисления. Режим доступа:
<http://ena.lp.edu.ua:8080/bitstream/ntb/10697/1/21.pdf> (дата обращения 09.10.2012).
16. Алфимцев А.Н., Девятков В.В. Интеллектуальные мультимодальные интерфейсы. Калуга: Полиграф-Информ, 2011. 328 с.
17. Иванова Г.С. Методология и средства разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники : автореф. дис. ... докт. техн. наук. М., 2007. 32 с.

Fuzzy process-oriented approach to nondeterministic design of intelligent multimodal interfaces

11, November 2012

DOI: 10.7463/1112.0495687

Alfimcev A.N.

Russia, Bauman Moscow State Technical University

alfim@bmstu.ru

Intelligent multimodal interface is user interface that can provide users with the opportunity to interact adequately to their modalities, recognize different situations to achieve users' goals, create a model of user's behavior and the environment in which it operates, modify behavior in accordance with the current state of the user and the environment. Design of intelligent multimodal interfaces is associated with overcoming a number of challenges and uncertainties by the developer. Therefore, they actively develop and study formal design methodology of intelligent multimodal interfaces which include a hierarchical set of interrelated formal models and operations on them, which can solve all tasks and related design automation. In this paper, the process-oriented approach to the formal non-deterministic design of intelligent multimodal interfaces using the concept of fuzzy running processes is presented. The method of selecting version of the draft of interface is considered. The language for visual representation of processes for the development of intelligent multimodal interfaces is described.

Publications with keywords:[design](#), [fuzzy logic](#), [intelligent multimodal interface](#), [nondeterminism](#), [theory of processes](#)

Publications with words:[design](#), [fuzzy logic](#), [intelligent multimodal interface](#), [nondeterminism](#), [theory of processes](#)

References

1. Grif M.G. *Avtomatizatsiia proektirovaniia protsessov funktsionirovaniia cheloveko-mashinnykh sistem po veroiatnostnym i nechetkim pokazateliyam. Dokt. diss.* [Design automation of processes of functioning of human-machine systems on probabilistic and fuzzy parameters. Dr. diss.]. Novosibirsk, 2002. 270 p.
2. Gribova V.V. *Avtomatizatsiia proektirovaniia, realizatsii i soprovozhdeniia pol'zovatel'skogo interfeisa na osnove ontologicheskogo podkhoda. Dokt. diss.* [Computer-aided design, implementation and support of user interface based on the ontological approach. Dr. diss.]. Vladivostok, 2007. 393 p.

3. Alfimtsev A.N., Deviatkov V.V. Printsipy formalizatsii metodologii determinirovannogo proektirovaniia intellektual'nogo mul'timodal'nogo interfeisa [Principles of formalization of methodology of deterministic design of intelligent multimodal interface]. *Vestnik MGTU im. N.E. Baumana. Ser. Priborostroenie* [Herald of the Bauman MSTU. Ser. Instrument Engineering], 2012, Spec. iss. no. 4 “*Komp'iuternye sistemy i tekhnologii*” [Computer systems and technologies], pp. 160-176.
4. Kleene S.C., Shannon C.E., McCarthy J., Ashby W.R. Representation of Events in Nerve Nets and Finite Automata. *Automata Studies*. Princeton University Press, 1956, pt. 1, pp. 3–42.
5. Morris A., Ulieru M. FRIEND: A Human-Aware BDI Agent Architecture. *Proc. of the International Conference «Systems, Man and Cybernetics»*. Anchorage, IEEE, 2011, pp. 2413-2418.
6. Bolie J., Cardella M., Blanvalet S., Juric M., Carey S., Chandran P., Coene Y., Geminiuc K., Zirn M., Gaur H. *BPEL Cookbook: Best Practices for SOA-based integration and composite applications development*. Birmingham-Mumbai: Pact publishing, 2006. 185 p.
7. Glass R.L. *Facts and Fallacies of Software Engineering*. New York, Addison-Wesley, 2004. (Russ. ed.: Glass R. *Fakty i zabluzhdeniia professional'nogo programmirovaniia*. St. Petersburg, Simvol-Plus, 2007. 240 p.).
8. Brooks Jr. F.P. *The Design of Design: Essays from a Computer Scientist*. New York, Addison-Wesley, 2010. 400 p.
9. Glass R.L. *Building Quality Software*. New Jersey, Prentice Hall, 1992. 369 p.
10. McConnell S. *Code Complete*. Redmond, Microsoft Press, 2004. (Russ. ed.: Makkonnell C. *Sovershennyi kod. Master-klass*. Moscow, Izdatel'sko-torgovyi dom «Russkaia Redaktsiia»; St. Petersburg, Piter, 2005. 896 s.).
11. Sutherland J. *Managing the iterative process*. Available at: <http://www.standishgroup.com/> , accessed 09.10.2012.
12. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 1975, no. 8, pp. 199-249.
13. Latypov I. *Psikhologiiia vybora* [The psychology of choice]. Available at: <http://www.gptc.ru/index.php/publications/124-psychology-of-choice.html>, accessed 09.10.2012.
14. Deviatkov V.V. Tekhnologiiia avtomatizirovannogo proektirovaniia sistem programmno-logicheskogo upravleniia [Technology of computer-aided design of systems of software-logic control]. *Avtomatizatsiia proektirovaniia: sb. nauch. tr.* [Computer-aided design: collection of scientific works]. Moscow, 1990, pp. 4-28.
15. Iliushko V., Kalmykov A. *Formalizatsiia biznes-protsessov na osnove apparatov RSA i π -ischisleniia* [Formalization of business processes on the basis of the apparatus of the RSA and π -calculus]. Available at: <http://ena.lp.edu.ua:8080/bitstream/ntb/10697/1/21.pdf> , accessed 09.10.2012.
16. Alfimtsev A.N., Deviatkov V.V. *Intellektual'nye mul'timodal'nye interfeisy* [Intelligent multi-modal interfaces]. Kaluga, Poligraf-Inform, 2011. 328 p.

17. Ivanova G.S. *Metodologiya i sredstva razrabotki algoritmov resheniya zadach analiza i sinteza struktur programmnoho obespecheniya i ustroystv vychislitel'noi tekhniki. Avtoreferat dokt. diss.* [Methodology and tools for developing algorithms for solving problems of analysis and synthesis of structures of software and devices of computer engineering. Abstract of dr. diss.]. Moscow, 2007. 32 p.