

А.Н. Алфимцев, В.В. Девятков

ПРИНЦИПЫ ФОРМАЛИЗАЦИИ МЕТОДОЛОГИИ ДЕТЕРМИНИРОВАННОГО ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНОГО МУЛЬТИМОДАЛЬНОГО ИНТЕРФЕЙСА

Эта проблема требует решения ряда непростых задач. Предложен новый подход к решению задач проектирования интеллектуальных мультимодальных интерфейсов, базирующийся на алгебре процессов, развиваемой на протяжении многих лет различными коллективами исследователей.

E-mail: alfim@bmstu.ru, deviatkov@iu3.bmstu.ru

Ключевые слова: интеллектуальный мультимодальный интерфейс, процесс, регулярные выражения, операторные схемы параллельных алгоритмов, процессные исчисления.

Интеллектуальный мультимодальный интерфейс (ИМИ) – пользовательский интерфейс, предоставляющий пользователям возможность взаимодействия адекватно их модальностям, распознавать различные ситуации для достижения своих целей, создавать модели поведения пользователей и среды, в которой он функционирует, модифицировать модели поведения в соответствии с текущим состоянием пользователей и среды [1].

Создание ИМИ – непростая задача. При этом актуальным является формализация процесса создания, которую можно взять за основу автоматизированного проектирования ИМИ. Любая формализация предполагает использование формальных моделей. В теории и практике искусственного интеллекта разработан ряд методологий, которые рассчитаны именно на интеллектуальное взаимодействие [2] – это мультиагентные системы (МАС) [3–8], в которых проектированию взаимодействия агентов отводится основная роль. В частности, в методологии работы [4] используется язык моделирования UML. В работе [9], посвященной методологии проектирования ИМИ, также используется язык UML. Все эти методологии не являются формализованными.

В связи с возрастающей сложностью интеллектуальных мультимодальных интерфейсов активно развиваются автоматизированные методологии проектирования интерфейсов [11–13], базирующиеся на использовании формальных моделей проектирования. В основу автоматизации проектирования процессов функционирования человеко-машинных систем работы [11] положены вероятностные и нечет-

кие показатели. В диссертации [12] автоматизация проектирования, реализации и сопровождения пользовательского интерфейса основана на онтологическом подходе. В работе [13] представлена методология разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники. Кроме работ, направленных на формализацию проектирования процессов программного обеспечения, в настоящее время активно развиваются методы формализации проектирования бизнес-процессов в различных областях применения [14], которые также могут быть использованы для создания формализованной методологии проектирования ИМИ.

Общими недостатками всех этих методологий является ориентация, как правило, на эрагатическую составляющую интерфейса и на одну формальную модель, что существенно ограничивает возможности методологии. По мнению авторов статьи, 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, приведенный в работе [9]. 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.

В настоящей работе рассматривается процессный подход к созданию таких формальных моделей в зависимости от решаемой задачи и принципы его использования для формализации методологии проектирования ИМИ, предложенной в работе [9].

Формальные процессные модели

Последовательный процесс. С каждым последовательным 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, 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. $a^* = \langle a_0 a_1 a_2 \dots a_{m-2} a_m \rangle$. *Выполнение нити* – 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*.

Последовательностным процессом (в дальнейшем просто *процессом*) P называется множество нитей $Th(P)$, которые он может выполнить. Воспользуемся для описания множества нитей $Th(P)$ языком регулярных выражений [15].

Если нить $a^* = \langle a_0 a_1 a_2 \dots a_{m-2} a_m \rangle$ конечна, то ее описание в языке регулярных выражений 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; 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 их можно представить 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'' закликивается, то для его описания будем использовать регулярное выражение $\{P''\}$. Если закликивание 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 $\{P'''\} = (\{P' P^a\} | \{P' P^b\} \dots | \{P' P^t\})$.

Поведение процесса P – правила выполнения нитей, принадлежащих множеству $Th(P)$. Эти правила для множества нитей, описанных на языке регулярных выражений, состоят в выполнении в каждый момент времени только одного действия, начиная с пустого действия, в направлении слева направо, или иначе, не используя понятия времени, действия в последовательностном процессе могут выполняться только поочередно слева направо в порядке записи их в нитях. Параллельно они выполняться не могут. Все подпроцессы последовательностного процесса используют один и тот же алфавит восприятий и реакций.

Последовательное выполнение действий. Если есть несколько последовательностных процессов P^1, P^2, \dots, P^t , то последовательное выполнение действий этими процессами определяется следующими процессными выражениями:

$$P \triangleq (P^1 \parallel P^2 \dots \parallel P^{t-1} \parallel P^t);$$

$$P^1 = !a_1 P_1^1;$$

$$P^2 = ?a_1 !a_2 P^2;$$

...

$$P^{t-1} = ?a_{t-2} !a_t P^t;$$

$$P^t = ?a_t P_1^t.$$

Первое процессное выражение задает процесс P , состоящий из параллельно выполняемых процессов P^1, P^2, \dots, P^t , первый из которых имеет внешнюю реакцию $!a_1$, одновременно являющуюся восприятием $?a_1$ процесса P^2 , который в свою очередь имеет внешнюю реакцию $!a_2$ одновременно являющуюся восприятием $?a_2$ процесса P^2 и т. д. до процесса P^t . Таким образом, не смотря на то, что процессы P^1, P^2, \dots, P^t являются параллельно выполняемыми, поведение каждого из них (кроме первого процесса) состоит в последовательной обработке реакций предыдущего процесса.

Разветвление выполнения действия. Если есть несколько последовательностных процессов P^1, P^2, \dots, P^t , то разветвление выполнения действия, являющегося реакцией процесса P^1 на процессы P^1, P^2, \dots, P^t , записывается в виде следующих процессных выражений:

$$\begin{aligned}
P &\triangleq P' \parallel (P^1 \parallel P^2 \dots \parallel P^{t-1} \parallel P^t); \\
P' &= !a P'_1; \\
P^1 &= ?a P^1_1; \\
P^2 &= ?a P^2_1; \\
&\dots \\
P^{t-1} &= ?a P^{t-1}_1; \\
P^t &= ?a P^t_1.
\end{aligned}$$

Первое процессное выражение задает процесс P , состоящий из параллельно выполняемых процессов P', P^1, P^2, \dots, P^t , первый из которых имеет внешнюю реакцию $!a$, а все остальные P^1, P^2, \dots, P^t в качестве первого восприятия имеют $?a$.

Синхронизация действий. Если есть несколько параллельно выполняемых последовательностных процессов P^1, P^2, \dots, P^t , каждый из которых имеет некоторую выходную реакцию, то синхронизация действий состоит в запуске некоторого процесса P' только после выполнения всех этих реакций. Синхронизация действий записывается в виде следующих процессных выражений:

$$\begin{aligned}
P &\triangleq (P^1 \parallel P^2 \dots \parallel P^{t-1} \parallel P^t) \parallel P'; \\
P^1 &= !a_1.P^1_1; \\
P^2 &= !a_2.P^2_1; \\
&\dots \\
P^t &= !a_t.P^t_1; \\
P' &= ?a_1. ?a_2. \dots . ?a_t.P''.
\end{aligned}$$

Процессы P^1, P^2, \dots, P^t имеют соответственно выходные реакции $!a_1, !a_2, \dots, !a_t$, которые синхронизируются процессом $P' = ?a_1. ?a_2. \dots . ?a_t.P''$.

Слияние выполнения действий. Если есть несколько альтернативно выполняемых последовательностных процессов P^1, P^2, \dots, P^t каждый из которых имеет одну и ту же выходную реакцию, то слияние выполнения действий состоит в запуске некоторого процесса P' только после выполнения одной из этих реакций. Слияние выполнения действий записывается в виде следующих процессных выражений:

$$\begin{aligned}
P''' &\triangleq (P^a | P^b \dots | P^t); \\
P &\triangleq (P^1 | P^2 \dots | P^t) || P'; \\
P^1 &= !a . P_1^1; \\
P^2 &= !a . P_1^2; \\
&\dots \\
P^t &= !a . P_1^t; \\
P' &= ? a . P'' .
\end{aligned}$$

Альтернативно выполняемые процессы P^1, P^2, \dots, P^t имеют выходную реакцию с одним и тем же именем $!a$, которые являются восприятием процесса $P' = ? a . P''$.

Альтернативное выполнение действий. Если есть несколько альтернативно выполняемых последовательностных процессов P^1, P^2, \dots, P^t , каждый из которых имеет соответственно свою входную реакцию $?a_1, ?a_2, \dots, ?a_t$, отличную от всех других входных реакций этих процессов, то альтернативное выполнение действий состоит в запуске некоторого процесса P' , состоящего из альтернативных подпроцессов, каждый из которых имеет соответственно одну их выходных реакций $!a_1, !a_2, \dots, !a_t$. Альтернативное выполнение действий записывается в виде следующих процессных выражений:

$$\begin{aligned}
P''' &\triangleq (P^a | P^b \dots | P^t); \\
P &\triangleq P' || (P^1 | P^2 \dots | P^t); \\
P' &= !a_1 . P_1^1 | !a_2 . P_2^1 | \dots | !a_t . P_t^1; \\
P^1 &= ? a_1 . P_1^1; \\
P^2 &= ? a_2 . P_1^2; \\
&\dots \\
P^t &= ? a_t . P_1^t .
\end{aligned}$$

Альтернативно выполняемые процессы $!a_1 . P | !a_2 . P_2^1 | \dots | !a_t . P_t^1$ имеют выходные реакции, которые являются соответственно восприятиями процессов $P^1 | P^2 \dots | P^t$.

До сих пор оперировали только непараметризованными процессами и действиями (только их имена). Для решения практических задач этого недостаточно. Поэтому наряду с непараметризованными

процессами и действиями используют также параметризованные, указывая в скобках после имени процесса или действия соответствующие параметры. Кроме того, в настоящей статье ограничимся только четкими процессами и действиями. Для решения задач создания формализованной методологии проектирования мультимодального интерфейса необходимо использовать нечеткие или вероятностные модели процессов. Вследствие ограниченности статьи эти модели не приводятся. На простом примере рассмотрим принципы использования приведенных четких процессных моделей для формализации проектирования ИМИ.

Формализация методологии проектирования ИМИ

В работе [9] приведен пример использования неформализованной методологии проектирования интеллектуального интерфейса для торгового киоска. В этом примере для описания процесса проектирования используется язык UML. Рассмотрим для этого примера принципы формализации проектирования интерфейса на основе представленных выше процессных моделей. При изложении принципов будем следовать этапам неформальной методологии проектирования интерфейса для этого примера, изложенной в книге [9].

Постановка задачи. Напомним, что мир киоска включает следующие субъекты: администратор, поставщики, продавцы, товары, покупатели, журналы продаж, цены, склад, выручка, прибыль. Поведение мира киоска состоит в следующем. Администратор закупает у поставщиков товары. Закупленные товары помещаются на склад. Продавец (в киоске всегда работает один продавец из категории всех продавцов) продает товар, получает за него деньги от покупателей и выдает товары со склада. При этом продавец фиксирует все продажи товара в журнале продаж, который он сдает в конце рабочего дня администратору киоска вместе с выручкой. Прибыль от работы киоска формируется за счет разницы между закупочными и продажными ценами товаров.

Задача разработки ИМИ для мира киоска в общем виде может быть сформулирована следующим образом: разработать такой ИМИ киоска, чтобы информация о состоянии всех субъектов киоска (продавцов, товаров, покупателей, журналов продаж, цен, склада, выручки, прибыли) была доступна администратору в любое время в печатном или электронном виде. Формулировка постановки задачи в таком виде носит самый общий характер и требует детализации в последующем.

Формирование начальных процессов. После выбора словаря имен субъектов и используя постановку задачи, формируется

начальная иерархия процессов разрабатываемого ИМИ. Выбор количества начальных процессных выражений и глубины их вложения субъективен и зависит как от разработчика, так и от заказчика. Все эти процессы формулируются как параллельные. По существу, они представляют иерархию целей, которые необходимо выполнить для разработки требуемого заказчиком ИМИ в терминах процессов безотносительно, в каком порядке они будут выполняться. Для нашего примера ограничимся начальной иерархией процессов, представленной в таблице.

Эта иерархия процессов представляется в виде дерева. Процессы нумеруют в десятичной системе. Сначала нумеруют подряд все процессы, не имеющие подпроцессов (подпроцессы 1-го уровня), затем для каждого процесса 1-го уровня после его номера и точки нумеруются подряд все его подпроцессы (подпроцессы процессов 1-го уровня), затем по аналогии подпроцессы 2-го уровня, 3-го уровня и т. д. В качестве начального выбран единственный процесс – разработать интеллектуальный интерфейс киоска – являющийся краткой формулировкой постановки задачи. Присвоим ему номер 1. Остальные подпроцессы и их имена указаны в первом столбце таблицы. Во втором столбце этой таблицы приведена содержательная интерпретация поведения процессов первого столбца.

Содержательная интерпретация процессов мира киоска

| Имя процесса и подпроцесса | Содержательная интерпретация поведения |
|----------------------------|--|
| P_1 | Разработать ИМИ киоска |
| $P_{1.1}$ | Обеспечить редактирование информации о продавцах |
| $P_{1.1.1}$ | Обеспечить возможность просмотра информации о продавцах, работающих в киоске |
| $P_{1.1.2}$ | Обеспечить возможность ввода информации о вновь принятых продавцах |
| $P_{1.1.3}$ | Обеспечить возможность удаления информации об уволенных продавцах |
| $P_{1.2}$ | Обеспечить ведение и редактирование информации о товарах на складе |
| $P_{1.2.1}$ | Обеспечить просмотр списка товаров, хранящихся на складе |
| $P_{1.2.2}$ | Обеспечить нахождение товаров по названиям в списке всех товаров |
| $P_{1.2.3}$ | Обеспечить ввод информации о товарах |
| $P_{1.2.4}$ | Обеспечить удаление информации о товарах |
| $P_{1.3}$ | Обеспечить ведение журналов продаж |
| $P_{1.3.1}$ | Обеспечить возможность ввода в текущий журнал продаж информации о продажах товаров и фактической выручке, которую сдает продавец за день |

| Имя процесса и подпроцесса | Содержательная интерпретация поведения |
|-----------------------------|--|
| <i>P</i> _{1.3.2} | Обеспечить возможность просмотра всех журналов продаж по дате создания и продавцу |
| <i>P</i> _{1.3.3} | Обеспечить возможность просмотра по всем журналам продаж информации о продажах товаров |
| <i>P</i> _{1.3.4} | Обеспечить возможность подготовки по журналам продаж отчетов прихода, расхода и выручки за отчетный период |
| <i>P</i> _{1.3.4.1} | Предусмотреть в отчете прихода наличие следующей информации о каждом товаре, поступившем за отчетный период: номер журнала продаж; дата поступления товара; код товара; наименование товара; количество поступившего товара; закупочная цена за единицу товара; сумма затрат на закупку всех единиц товара данного вида; сумма всех затрат на закупку всех товаров |
| <i>P</i> _{1.3.4.2} | Обеспечить возможность просмотра информации в формате отчета прихода по наименованию товара или ключевым словам, входящим в его наименование |
| <i>P</i> _{1.3.4.3} | Предусмотреть в отчете расхода наличие следующей информации о каждом товаре, поступившем за отчетный период: номер журнала продаж; дата продажи; продавец; наименование проданного товара; количество товара, которое имелось в наличии до продажи; проданное количество товара; количество товара, которое осталось в наличии; цена, по которой продан товар; сумма продажи товара; закупочная цена товара на момент продажи; полученная прибыль от продажи товара данного вида; суммарная прибыль, полученная от продажи товаров, попавших в отчет |
| <i>P</i> _{1.3.4.5} | Обеспечить возможность просмотра информации в формате отчета расхода по наименованию товара или ключевым словам, входящим в его наименование, и продавцу, продавшему товар |
| <i>P</i> _{1.3.4.6} | Предусмотреть в отчете выручки наличие следующей информации о каждом товаре, поступившем за отчетный период: номер журнала продаж; дата продажи; продавец; расчетная выручка; фактическая выручка; сумма недостачи; суммарная фактическая выручка |
| <i>P</i> _{1.3.4.7} | Обеспечить возможность просмотра информации в формате отчета расхода по продавцу, сдавшему журнал |

Начальные процессные выражения мира киоска приведены ниже:

$$\begin{aligned}
P &\triangleq P_1(P^1 \parallel P^2 \parallel P^3), \\
P^1 &\triangleq P_{1.1}(P_{1.1.1} \parallel P_{1.1.2} \parallel P_{1.1.3}), \\
P^2 &\triangleq P_{1.2}(P_{1.2.1} \parallel P_{1.2.2} \parallel P_{1.2.3} \parallel P_{1.2.4}), \\
P^3 &\triangleq P_{1.3}(P_{1.3.1} \parallel P_{1.3.2} \parallel P_{1.3.3} \parallel P^4), \\
P^4 &\triangleq P_{1.3.4}(P_{1.3.4.1} \parallel P_{1.3.4.2} \parallel P_{1.3.4.3} \parallel P_{1.3.4.4}).
\end{aligned}$$

Структуризация начальных процессов. На этапе структуризации начальных процессов осуществляется, если это необходимо, дальнейшая их детализация. Например, процесс 1.2.3 «Обеспечить ввод информации о товарах» может быть разделен на два альтернативных подпроцесса «Ввести информацию об имеющемся на складе товаре» и «Ввести информацию о новом товаре». В результате получим процессное выражение:

$$P_{1.2.3} = P_{1.2.3.1} | P_{1.2.3.2}.$$

Формирование взаимодействия процессов. Для описания взаимодействия процессов вводятся названия действий и, используя эти действия, начальные процессы преобразуются во взаимодействующие процессы, которые в следующем разделе будут распределены между агентами. Процессные выражения для взаимодействующих процессов *Редактирование*, *Просмотр*, *Поиск*, *Удаление*, *Ввод* показаны далее. Для их описания использованы действия *просмотреть*, *найти*, *вести*, *удалить*, *просмотрено*, *найденно*, *введено*, *удалено*, *новый товар*, *имеющийся товар*:

$$P^2 \triangleq P_{1.2}(P_{1.2.1} \parallel P_{1.2.2} \parallel P_{1.2.3} \parallel P_{1.2.4}),$$

$$P_{1.2} \triangleq ! \text{редактировать.} \text{Редактирование,}$$

$$\text{Редактирование} \triangleq ! \text{просмотреть.} ! \text{подготовка.} ! \text{вывод.}$$

$$! \text{окончание.} 0. | ! \text{найти.} 0. | ! \text{вести.} 0. | ! \text{удалить.} 0.$$

$$P_{1.2.1} \triangleq ? \text{просмотреть.} \text{Просмотр,}$$

$$P_{1.2.2} \triangleq ? \text{найти.} \text{Поиск,}$$

$$P_{1.2.3} \triangleq ? \text{вести.} \text{Ввод,}$$

$$\text{Ввод} \triangleq P_{1.2.3.1} | P_{1.2.3.2},$$

$$P_{1.2.3.1} \triangleq ? \text{новый_товар.} \text{Ввод_нового_товара,}$$

$$P_{1.2.3.2} \triangleq ? \text{имеющийся_товар.} \text{Ввод_имеющегося_товара,}$$

$$P_{1.2.4} \triangleq ? \text{удалить.} \text{Удаление.}$$

Выделение взаимодействующих агентов. Для выделения агентов взаимодействующие процессы разбиваются на группы, каждая из которых соответствует одному агенту, выполняющему эти процессы. Для нашего примера выделим трех агентов: администратора, оператора и кладовщика. Администратору поручается выполнять процесс *Редактирование*, оператору – процесс *Просмотр*, кладовщику – процессы *Поиск*, *Ввод*, *Удаление*. Агенты являются процессами *Администратор*, *Оператор*, *Кладовщик*. Процессные выражения, задающие результат введения агентов и их взаимодействие, показаны ниже:

$$P^2 \triangleq P_{1,2}(P_{1,2,1} \parallel P_{1,2,2} \parallel P_{1,2,3} \parallel P_{1,2,4}),$$

$$\text{Администратор} \triangleq P_{1,2} = ? \text{редактировать.Редактирование},$$

$$\text{Редактирование} \triangleq !\text{просмотреть.}! \text{подготовка.}! \text{вывод.}$$

$$! \text{окончание.0.} | ! \text{найти.0.} | ! \text{ввести.0.} | ! \text{удалить.0.}$$

$$\text{Оператор} \triangleq P_{1,2,1},$$

$$P_{1,2,1} \triangleq ? \text{просмотреть.Просмотр},$$

$$\text{Кладовщик} \triangleq P_{1,2,2} \parallel P_{1,2,3} \parallel P_{1,2,4},$$

$$P_{1,2,2} \triangleq ? \text{найти.Поиск},$$

$$P_{1,2,3} \triangleq ? \text{ввести.Ввод.}$$

$$\text{Ввод} \triangleq P_{1,2,3,1} | P_{1,2,3,2},$$

$$P_{1,2,3,1} \triangleq ? \text{новый_товар.Ввод_нового_товара},$$

$$P_{1,2,3,2} \triangleq ? \text{имеющийся_товар.Ввод_имеющегося_товара},$$

$$P_{1,2,4} \triangleq ? \text{удалить.Удаление.}$$

Выделение интерфейсов агентов. После выделения агентов и назначения им процессов, которые они выполняют, внутри этих процессов вводятся интерфейсы, если это необходимо. Для нашего примера для всех процессов *Просмотр*, *Поиск*, *Удаление*, *Ввод_нового_товара*, *Ввод_имеющегося_товара* вводятся соответствующие интерфейсы: *Интерфейс просмотра*, *Интерфейс поиска*, *Интерфейс удаления*, *Интерфейс ввода нового товара*, *Интерфейс ввода имеющегося товара*. Процессные выражения, задающие результат выделения интерфейсов, приведены ниже:

$$P^2 \triangleq P_{1,2}(P_{1,2,1} \parallel P_{1,2,2} \parallel P_{1,2,3} \parallel P_{1,2,4}),$$

$$\text{Администратор} \triangleq P_{1,2} = ? \text{редактировать.Редактирование},$$

$$\text{Редактирование} \triangleq !\text{просмотреть.}! \text{подготовка.}! \text{вывод.}!$$

$$\text{окончание0.} | ! \text{найти.0.} | ! \text{ввести.0.} | ! \text{удалить.0.}$$

$$\text{Оператор} \triangleq P_{1,2,1},$$

$$P_{1,2,1} \triangleq ? \text{просмотреть.Просмотр},$$

$$\text{Просмотр} \triangleq \text{Интерфейс_просмотра.}! \text{просмотрено.0.}$$

Кладовщик $\hat{=}$ $P_{1.2.2} \parallel P_{1.2.3} \parallel P_{1.2.4}$,
 $P_{1.2.2} \hat{=}$? найти.Поиск,
Поиск $\hat{=}$ Интерфейс_поиска.!найдено.0.
 $P_{1.2.3} \hat{=}$? ввести.Ввод,
Ввод $\hat{=}$ $P_{1.2.3.1} | P_{1.2.3.2}$,
 $P_{1.2.3.1} \hat{=}$? новый_товар.Ввод_нового_товара,
 $P_{1.2.3.2} \hat{=}$? имеющийся_товар.Ввод_имеющегося_товара,
Ввод_нового_товара $\hat{=}$
? новый_товар.Интерфейс_ввода_нового_товара.!введено.0.
Ввод_имеющегося_товара $\hat{=}$
? имеющийся_товар.Интерфейс_ввода_имеющегося_товара.!введено.0.
 $P_{1.2.4} \hat{=}$? удалить.Удаление,
Удаление $\hat{=}$ Интерфейс_удаления.!удалено.0.

Формирование протоколов интерфейсов агентов. Протокол интерфейса агента – процесс, состоящий из восприятий, реакций и подпроцессов. Восприятия и реакции этого процесса являются соответственно реакциями и восприятиями других агентов, что, собственно, и составляет интерфейс. Протокол интерфейса агента-оператора при выполнении им процесса *Просмотр*, который на наш взгляд не требует пояснения в силу своей очевидности, показан далее:

$P^2 \hat{=}$ $P_{1.2}(P_{1.2.1} \parallel P_{1.2.2} \parallel P_{1.2.3} \parallel P_{1.2.4})$,
Администратор $\hat{=}$ $P_{1.2} = ?$ редактировать.Редактирование,
Редактирование $\hat{=}$!просмотреть.!подготовка.!вывод.
!окончание.|!найти.0.|!ввести.0.|!удалить.0.

Оператор $\hat{=}$ $P_{1.2.1}$,
 $P_{1.2.1} \hat{=}$? просмотреть.Просмотр,
Просмотр $\hat{=}$ Интерфейс_просмотра.!просмотрено.0.

Кладовщик $\hat{=}$ $P_{1.2.2} \parallel P_{1.2.3} \parallel P_{1.2.4}$,
 $P_{1.2.2} \hat{=}$? найти.Поиск,
Поиск $\hat{=}$ Интерфейс_поиска.!найдено.0.
 $P_{1.2.3} \hat{=}$? ввести.Ввод.
Ввод $\hat{=}$ $P_{1.2.3.1} | P_{1.2.3.2}$,
 $P_{1.2.3.1} \hat{=}$? новый_товар.Ввод_нового_товара,
 $P_{1.2.3.2} \hat{=}$? имеющийся_товар.Ввод_имеющегося_товара,
Ввод_нового_товара $\hat{=}$
? новый_товар.Интерфейс_ввода_нового_товара.!введено.0.
Ввод_имеющегося_товара $\hat{=}$
? имеющийся_товар.Интерфейс_ввода_имеющегося_товара.!введено.0.
 $P_{1.2.4} \hat{=}$? удалить.Удаление,
Удаление $\hat{=}$ Интерфейс_удаления.!удалено.0,

Интерфейс _ просмотра \triangleq *Подготовка _ к _ просмотру.*
Упорядочивание _ списка _ товаров.
Вывод _ списка _ товаров.
Окончание _ просмотра.0.

Подготовка _ к _ просмотру \triangleq ? *подготовка.*
!список _ товаров _ упорядочен.0.

Вывод _ списка _ товаров \triangleq ? *вывод.*
!список _ товаров _ выведен.0.

Окончание _ просмотра \triangleq ? *окончание.*
!список _ товаров _ просмотрен.0.

Проверка правильности функционирования ИМИ

Проверка правильности функционирования ИМИ – важная задача, особенно это актуально на ранних этапах проектирования. Процессный подход представляет для этого несколько возможностей. Одна из них – использование механизма редукции (вывода) на основе редукционных правил [16]. Суть этого механизма применительно к моделям процессов ИМИ состоит в том, что выбираются группы агентов, правильность интерфейса между которыми необходимо проверить и, используя правила редукции, начинают выполнять интерфейс. Если результаты интерфейса достигают цели, то тем самым доказывается правильность функционирования. Также можно проверять различные свойства ИМИ [17]. Таким образом, метод проверки на правильность функционирования ИМИ состоит из следующих шагов:

- 1) выбор группы процессов;
- 2) выбор цели проверки;
- 3) выполнение перебора всех процессов в группе;
- 4) если есть возможность применения редукции к процессам, то выполнить редукцию. В противном случае переход к шагу 6;
- 5) если цель достигнута, то переход к шагу 7. В противном случае к шагу 3;
- 6) неуспешное завершение проверки правильности функционирования ИМИ;
- 7) успешное завершение проверки правильности функционирования ИМИ.

Для нашего примера, начиная с восприятия администратора ?*редактировать*, можно успешно достичь реакции оператора !*просмотрено*. Несколько последовательных шагов редукции показаны ниже:

Администратор $\hat{=} P_{1,2} =$ Редактирование,

Редактирование $\hat{=} !$ просмотреть! $!$ подготовка! $!$ список_товаров_упорядочен! $!$ вывод.

$!$ список_товаров_выведен! $!$ окончание.список_товаров_просмотрен|

| $!$ найти.0. $!$ ввести.0. $!$ удалить.0.

Оператор $\hat{=} P_{1,2,1}$,

$P_{1,2,1} \hat{=} ?$ просмотреть.Просмотр,

Просмотр $\hat{=} \text{Интерфейс_просмотра}!$ просмотрено.0.

Интерфейс_просмотра $\hat{=} \text{Подготовка_к_просмотру}$.

Упорядочивание_списка_товаров.

Вывод_списка_товаров.

Окончание_просмотра.0.

Подготовка_к_просмотру $\hat{=} ?$ подготовка.

$!$ список_товаров_упорядочен.0.

Вывод_списка_товаров $\hat{=} ?$ вывод.

$!$ список_товаров_выведен.0.

Окончание_просмотра $\hat{=} ?$ окончание! $!$ 0.

Администратор $\hat{=} P_{1,2} = !$ подготовка! $!$ вывод.

$!$ окончание. $!$ найти.0. $!$ ввести.0. $!$ удалить.0.

Оператор $\hat{=} \text{Подготовка_к_просмотру}$.

Упорядочивание_списка_товаров.

Вывод_списка_товаров.

Окончание_просмотра! $!$ просмотрено.0.

Подготовка_к_просмотру $\hat{=} ?$ подготовка.

$!$ список_товаров_упорядочен.0.

Вывод_списка_товаров $\hat{=} ?$ вывод.

$!$ список_товаров_выведен.0.

Окончание_просмотра $\hat{=} ?$ окончание.

$!$ список_товаров_просмотрен.0.

Администратор $\hat{=} P_{1,2} = !$ вывод! $!$ окончание. $!$ найти.0. $!$ ввести.0. $!$ удалить.0.

Оператор $\hat{=} \text{Упорядочивание_списка_товаров}$.

Вывод_списка_товаров.

Окончание_просмотра! $!$ просмотрено.0.

Вывод_списка_товаров $\hat{=} ?$ вывод.

$!$ список_товаров_выведен.0.

Окончание_просмотра $\hat{=} ?$ окончание.

$!$ список_товаров_просмотрен.0.

В заключение следует отметить, что предложенные принципы новой методологии проектирования интеллектуальных интерфейсов на основе процессного подхода к созданию формальных моделей ИМИ позволяют:

- решать задачи оптимизации, определения функциональной правильности, проверки на синтаксическую и семантическую корректность, автоматической коррекции спроектированного процесса ИМИ на основе проверки эквивалентности с любым эталонным процессом;
- осуществлять автоматическую проверку свойств ИМИ;
- автоматизировать практически все этапы формального проектирования ИМИ;
- разрабатывать интеллектуальное программное обеспечение, в частности интеллектуальные мультимодальные интерфейсы.

СПИСОК ЛИТЕРАТУРЫ

1. Девятков В. В., Алфимцев А. Н. Нечеткая конечно-автоматная модель интеллектуального мультимодального интерфейса // Проблемы управления. – 2011. – № 2. – С. 69–77.
2. FIPA official web site [Электронный ресурс]. URL: <http://www.fipa.org/> (дата обращения: 31.05.2012).
3. Arias C. I. S., Daltrini B. M. A Multi-Agent Environment for User Interface Design // Proc. of the 22nd EUROMICRO Conference. – 1999. – P. 242.
4. DeLoach S. A. Multiagent Systems Engineering. A Methodology and Language for Designing Agent Systems// In Proc. of Agent Oriented Information Systems. – 1999. – P. 45–57.
5. Wooldridge M. J., Jennings N. R. and Kinny D. The Gaia methodology for agent-oriented analysis and design // Autonomous Agents and Multi-Agent Systems. September 2000. – No. 3(3). – P. 285–312.
6. Depke R., Heckel R. Formalizing the Development of Agent-Based Systems Using Graph Processes // In Proc. of the ICALP'2000 Satellite Workshops, Workshop on Graph Transformation and Visual Modelling Techniques (GTVMT'00). – 2000. – P. 419–426.
7. Aridor Y., Lange D. B. Agent Design Patterns: Elements of Agent Application Design // In Proc. of the second international conference on Autonomous agents, 1997. – P. 108–115.
8. Brazier, F. M. T., Jonker, C. M., Treur, J.: Principles of Component-Based Design of Intelligent Agents. Data and Knowledge Engineering 41, 2002. – P. 1–27.
9. Алфимцев А. Н., Девятков В. В. Интеллектуальные мультимодальные интерфейсы. – Калуга: ООО «Полиграф-Информ», 2011. – 328 с.
10. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS.: Пер. с англ.; под общей редакцией проф. С. Орлова. – СПб.: Питер, 2000. – 736 с.
11. Гриф М. Г. Автоматизация проектирования процессов функционирования человеко-машинных систем по вероятностным и нечетким показателям: Дис. ... д-ра техн. наук. Новосибирск, 2002.
12. Грибова В. В. Автоматизация проектирования, реализации и сопровождения пользовательского интерфейса на основе онтологического подхода: дис. ... д-ра техн. наук. Владивосток, 2007.
13. Иванова Г. С. Методология и средства разработки алгоритмов решения задач анализа и синтеза структур программного обеспечения и устройств вычислительной техники: автореферат дис. ... д-ра техн. наук. Москва, 2007.
14. Илюшко В., Калмыков А. Формализация бизнес-процессов на основе аппаратов PCA и π -исчисления // Компьютерные системы и информационные технологии. – 2011. – С. 144–152.

15. Kleene S. C. Representation of Events in Nerve Nets and Finite Automata. In Shannon, Claude E.; McCarthy, John. Automata Studies. Princeton University Press. – 1955. – P. 3–42.
16. Milner R. A Calculus of Communicating Systems. Springer-Verlag New York: Inc. Secaucus, NJ, 1980. – 171 p.
17. Алфимцев А. Н., Девятков В. В. Необходимые и достаточные формальные свойства мультимодального интерфейса // Вестник МГТУ им. Н.Э. Баумана Сер. Приборостроение. Спец. вып. «Информационные технологии и компьютерные системы». – 2011. – С. 159–167.

Статья поступила в редакцию 14.05.2012