

Таким образом, авторы представляют вывод во фреймовой сети при помощи процедур означивания и демонов, хотя существуют и другие способы вывода, например, с помощью указателей наследования [2]. Обозначена так же возможность использования в качестве значений слотов нечетких и лингвистических значений, в качестве процедур – мягких вычислений измерений и сопоставлений, в качестве мер близости – нечетких мер.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Минский М.* Фреймы для представления знаний: Пер. с англ. М.: Энергия, 1979. 159с.
2. *Х. Уэно, Т.Кояма, Т.Окамото, Б.Мацуби, М.Исидзука.* Представление и использование знаний. М.: Мир, 1989. 220с.
3. *Сергеев Н.Е.* Фреймовая реализация гибридных моделей принятия решений// Интеллектуальные САПР. Таганрог: Изд-во ТРТУ, 1992.
4. *Сергеев Н.Е.* Информационная модель многоуровневой АСУ ТП./ Известия ТРТУ. Таганрог: Изд-во ТРТУ, 2000.

УДК 519.68

**В.Л. Сахаров**

#### **АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К СОЗДАНИЮ СИСТЕМЫ ПОДДЕРЖКИ ДЕЯТЕЛЬНОСТИ ПРИ РАБОТЕ С АВТОМАТИЗИРОВАННЫМИ КОМПЬЮТЕРНЫМИ СИСТЕМАМИ В МЕДИЦИНСКИХ УЧРЕЖДЕНИЯХ**

В настоящее время повсеместное внедрение вычислительной техники позволило автоматизировать, и, как следствие, значительно облегчить труд медиков, работающих в разных сферах деятельности медицинских учреждений. К таким видам деятельности можно отнести как работу с распределенными базами данных по ведению электронных историй болезни, статистических отчетов, так и работу с автоматизированными компьютерными диагностическими комплексами. Однако, с появлением вычислительной техники у пользователей таких систем возникли дополнительные проблемы, которые, в основном, связаны со слабым знанием вычислительной техники и отсутствием опыта работы с программными системами. Таким образом, иногда имеет место ситуация, когда существенное сокращение временных затрат, связанное с автоматизацией расчетов и вычислений компенсируется длительными действиями пользователей по выбору нужных действий. И должно пройти значительное время, пока пользователи привыкают к выполнению действий по работе с программным обеспечением. В итоге автоматизация внедряется очень медленно, и, зачастую, с противодействиями со стороны пользователей.

Для упрощения работы и ускорения обучения программным средствам были разработаны и внедрены различные средства поддержки. Вначале разработчики ограничивались подробным бумажным или электронным описанием принципам работы с системой. Но изучение описаний и поиск в них нужных разделов требовало также достаточно много времени. В дополнение к описаниям стали разрабатываться интерактивные средства, называемые «хелпер» (от англ. help – помощь). Эти средства позволяют, находясь в каком либо окне программы дать текстовые

рекомендации по тем действиям, которые может сделать оператор. В настоящее время появились также и видео-ролики, демонстрирующие, куда и когда нужно нажимать. Но опять же работа с ними требует выполнения определенных действий, а в ситуациях, когда пользователь просто находится в тупике и не знает, что ему делать дальше, такие системы поддержки не смогут оказать явную помощь. Следующим современным этапом разработки систем поддержки стали так называемые «скрепки-помощники». Наиболее наглядно такой помощник реализован в Microsoft Office. Фактически, это уже агент, который активизируется при затруднительных ситуациях при работе с программами и подсказывает, что наиболее предпочтительно сделать в настоящий момент, а также периодически выдает полезные советы по использованию программного продукта. Он фактически является интеллектуальным посредником между пользователем и средой, в которой тот работает [Maes, 1994].

В настоящей работе предлагается разработка еще более интеллектуального посредника, который, помимо простых подсказок, при затруднительных ситуациях «вел» пользователя по верному пути, параллельно обучая его работе с системой. Введем понятие «агента-помощника» а:

$$a(N_1, N_2, \dots, N_n),$$

где  $N_i$  – набор атрибутов агента-помощника  $i = \overline{1, n}$ ;

Атрибутом в данном случае называются свойства агента, каждое из которых активизирует соответствующий блок процедур по организации интеллектуального взаимодействия с пользователем. Основными атрибутами в данном случае являются: свойство накопления знаний о действиях пользователя; свойство корректного и своевременного вызова; свойство выбора верного решения за пользователя; свойство ручной коррекции действий агента.

Представим далее схему взаимодействия атрибутов агента со средой использования (рис.1).

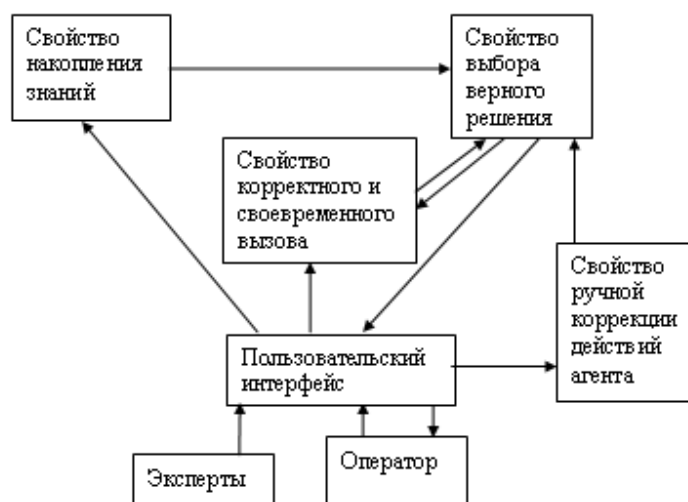


Рис.1

Опишем работу данной схемы. Взаимодействие оператора с системой и агентом-помощником осуществляется через пользовательский интерфейс программного обеспечения. На начальном этапе для корректной работы агента-помощника необходимо собрать знания либо от самого пользователя, либо от экспертов, которым на этапе разработки система была дана в пользование. Для каждого шага эксплуатации системы (фактически для каждого рабочего окна программного обеспечения) строится матрица накопления знаний  $E$  размером  $m \times n$ , отражающая частоту выполнения действий оператора. Столбцы этой матрицы  $X_i$  ( $i=1, \dots, m$ ) представляют собой цель использования данного рабочего окна, а строки  $Y_j$  ( $j=1, \dots, n$ ) – возможные пути действия оператора. Возможными путями действия считаются нажатия кнопок или выбор пунктов меню текущего окна. На пересечении столбцов  $X_i$  и строк  $Y_j$  находится число  $e[X_i, Y_j] \in [0,1]$ , показывающее коэффициент нечеткости при выборе действия оператора. Этот коэффициент нормируется к единице и строится на основе формулы

$$e[X_i, Y_j] = \left( \frac{k_{i,j}}{S_i} + M_i \right) / 2, \quad (1)$$

где  $k_{ij}$  – количество выборов  $j$ -го действия для  $i$ -го рабочего окна;  $S_i$  – общее количество выборов в течение процесса обучения агента;  $M_i \in [0,1]$  – коэффициент, задаваемый разработчиком и являющийся экспертной оценкой возможного выбора действия.

Здесь коэффициенты  $k$  и  $S$  подсчитываются агентом при работе с системой экспертов, уже имеющих опыт работы с ней. Таковыми могут быть либо медицинские идеологи разработки программного обеспечения, либо опытные пользователи, работающие с системой длительное время и хорошо знающие все ее возможности. Коэффициент  $M$  формируется и задается разработчиком с учетом идеи реализации того или иного рабочего окна программного обеспечения. За основу данный коэффициент брать не может в связи с тем, что разработчик не всегда адекватно оценивает реальную работу пользователя, но учитываться при формировании матрицы накопления знаний, естественно, не может. Таким образом, результатом работы агента-помощника в плане накопления знаний будет формирование множества подобных матриц равное количеству рабочих окон программного обеспечения, значения которых будут храниться, изменяться и добавляться в атрибуте «Свойства накопления знаний». Данная информация лежит в основе работы атрибута «Свойство выбора верного решения».

Перейдем к рассмотрению непосредственной работы агента-помощника при работе пользователя с системой. Есть два пути вызова помощника: первый – непосредственный вызов его через пункты меню или горячие клавиши; второй – автоматизированный вызов помощника после определенного интервала времени ожидания действий оператора. В реальных условиях первый вариант предпочтительнее, т.к. является более объективным. При использовании второго варианта необходимо задать массив  $A [1..N]$  временных интервалов ожидания действий оператора для каждого рабочего окна программного обеспечения. Здесь  $N$  – количество рабочих окон. Временные интервалы могут быть определены как на основе экспертной оценки на этапе создания агента, так и заданы вручную самим пользователем. Следующее действие этого атрибута – запрос и получение кода рабочего окна. Этот код необходим для передачи атрибуту «Свойство выбора верного решения».

При активации атрибута «Свойство корректного и своевременного вызова» он передает управление атрибуту «Свойство выбора верного решения» вместе с кодом рабочего окна. Работу этого атрибута можно проиллюстрировать схемой (рис.2).

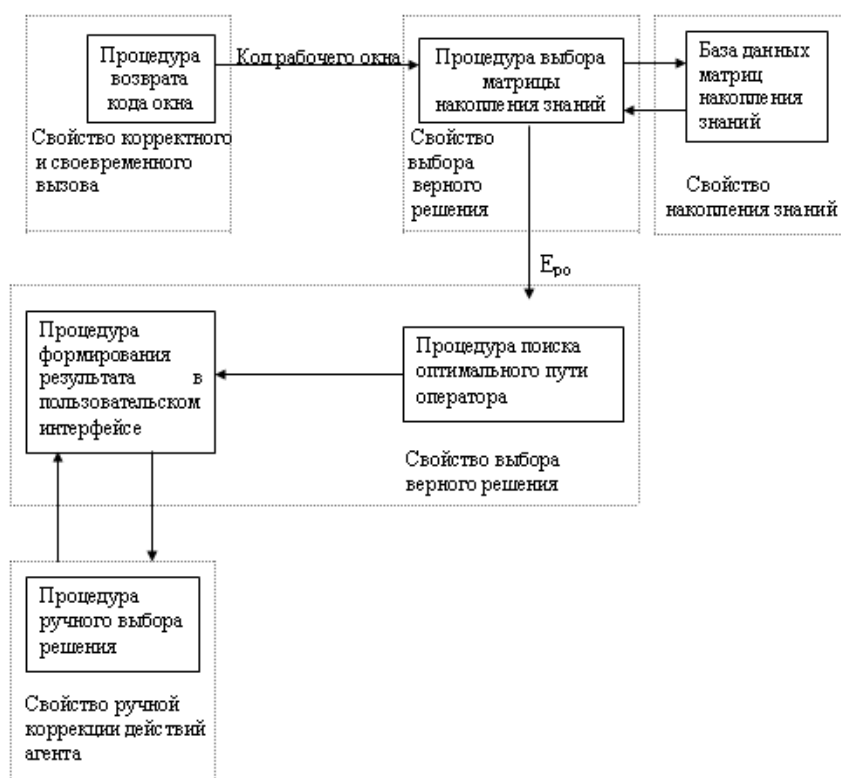


Рис.2

На данной схеме (см. рис.2) показано взаимодействие отдельных процедур атрибутов агента-помощника. Опишем это взаимодействие. При поступлении кода рабочего окна процедура выбора матрицы накопления знаний атрибута «Свойство выбора верного решения» обращается в базу данных, хранящую матрицы накопления знаний. Эта база принадлежит атрибуту «Свойство накопления знаний». Происходит выбор нужной матрицы накопления знаний. Данные этой матрицы передаются в процедуру поиска оптимального пути оператора атрибута «Свойство выбора верного решения». Поиск оптимального пути возможно выполнить простым перебором и сравнением  $e(X_i, Y_j)$  для всех  $j$  с учетом выбранного  $i$  и поиском максимального значения  $e$ . Индекс  $j$  у этого максимального значения будет соответствовать тому действию, которое агент-помощник считает наилучшим для данной ситуации. Далее вызывается процедура формирования результата в пользовательском интерфейсе. Эта процедура фактически представляет собой блок управления интерфейсом программного обеспечения и выполняет вместо пользователя действие по управлению. Однако, такой агент-помощник не защищен от ошибок и не-

корректных действий. А это фактически может привести к полной неработоспособности системы и неприятию ее пользователем. Поэтому агент-помощник использует еще один атрибут – «Свойство ручной коррекции действий агента». Этот атрибут включает в себя процедуру ручного выбора решения. Фактически, оператор на каждом шаге работы агента может скорректировать его работу, выбрав нужные действия. В случае частого возникновения ситуаций ручной коррекции, целесообразно при работе агента учитывать мнение пользователя. Для этого в формулу (1) необходимо внести соответствующие изменения. Теперь формула примет вид:

$$e[X_i, Y_j] = \left( \frac{k_{i,j}}{S_i} + \frac{n_{i,j}}{P_i} + M_i \right) / 3, \quad (2)$$

где  $n_{i,j}$  – количество выборов  $j$ -го действия при выполнении ручной коррекции;  $P_i$  – общее количество коррекций, которые были выполнены вручную.

Таким образом, в формуле (2) учитывается и мнение экспертов, и мнение разработчиков, и, если это необходимо, мнение самого пользователя. Можно предположить, что такой подход позволит наиболее корректно построить матрицы накопления знаний и организовать работу агента-помощника.

Итак, атрибуты агента-помощника представляют собой наборы процедур, программных объектов и баз данных, которые, взаимодействуя между собой, решают задачи поддержки работы оператора с медицинскими информационными системами. В качестве примера использования данный подход реализован практически при разработке компьютерной электромиографической системы «Феникс».

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. В.Б. Тарасов. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал УРСС, 2002. 352с.
2. Maes P. Agent That Reduces Work and information Overload // Communication of the ACM. 1994. Vol. 37
3. Sakharov V.L. The Principles of Networks Designing of Medical Establishment. Proceedings of 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002), 2002.

УДК 681.3.06

**В.Е. Подобедов**

#### **ОБНАРУЖЕНИЕ СЛОЖНЫХ СОБЫТИЙ В ПЛАТЕЖНОЙ СИСТЕМЕ С ПОМОЩЬЮ МОНИТОРИНГА АКТИВНОСТИ СУБЪЕКТОВ**

Для сложных событий, происходящих в платежной системе, не всегда очевиден даже факт их наступления. Для обнаружения таких событий приходится проводить наблюдение за активностью субъектов системы. Изменение характеристик активности субъекта позволяет предположить наступление связанного с этим субъектом события. Форма представления характеристик зависит от разнообразия выполняемых субъектом операций. Их следует представлять посредством видовых распределений.

В процессе функционирования большой платежной системы (например, российской платежной системы СТБ КАРД), обеспечивающей выполнение потока