P.303-304. **2.** *Hornik K., Stinchcombe M., White H.* Multilayer feedforward networks are universal approximators // Neural Networks.1989.№2.P.359-366. 3**.** *Hornik K.* Approximation capabilities of multiplayer feedforward networks // Neural Networks.1991.4.P.251-257. **4.** *Golden R.M.* Mathematical Methods for Neural Networks Analysis and Design.-Cambridge, Massachusetts: The MIT Press, 1996. 420p. **5.** *Kruschke J.K., Movellan S.R.* Benefits of gain: Speeded learning and minimal layers backpropagation networks// IEEE Trans. on Syst., Man and Cybernetics.1991.21. P.273-280. **6.** *Бодянский Е.В., Кулишова Н.Е., Руденко О.Г.* Обобщённый алгоритм обучения формального нейрона // Кибернетика и системный анализ. 2002. №6. С.176-182. **7.** *Бодянский Е.В., Кулишова Н.Е., Руденко О.Г.* Об одной модели формального нейрона // Доповіді НАН України. 2001. №4. С.69-73. **8.** *Bodyanskiy Ye., Chaplanov O., Kolodyazhniy V., Otto P.* Adaptive quadratic radial basis function networks for time series forecasting // Proc. East West Fuzzy Coll.-Zittau/Goerlitz: HS, 2002. P.164-172. **9.** *Shepherd A.J.* Second-Order Methods for Neural Networks.-London: Springer-Verlag, 1997. 145p. **10.** *Ham F.M., Kostanic I.* Principles of Neurocomputing for Science and Engineering. N.Y.:Mc Graw-Hill, Inc., 2001. 642p. **11.** *Goodwin G.C., Ramadge P.J., Caines P.E.* A globally convergent adaptive predictor // Automatica.-1981. №1. P.135-140.

Поступила в редколлегию 23.07.2003

Рецензент: д-р техн. наук, проф. Любчик Л.М.

Бодянский Евгений Владимирович, д-р техн. наук, профессор кафедры искусственного интеллекта, научный руководитель ПНИЛ АСУ ХНУРЭ, член IEEE, WSES. Научные интересы: нейро-фаззи-системы. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-890.

E-mail:Bodyanskiy@ieee.org, bodya@kture.kharkov.ua

Полушкина Наталия Александровна, ст. лаборант каф. филосифии ХНУРЭ. Научные интересы: искусственные нейронные сети, нейросетевое прогнозирование. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-465. E-mail: Natalie_p@mail.ru

УДК 519.5

ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПРОГРАММНОГО АГЕНТА

ФИЛАТОВ В.А.

Рассматривается подход к построению модели программного агента на основе фреймовой структуры. Обосновывается формальный подход, используемый в теории образов, к задаче проектирования логической структуры взаимосвязанных программных агентов. Предложенные модели могут использоваться для разработки мультиагентных систем администрирования информационных ресурсов распределенных вычислительных систем.

Введение

Исследования в области интеллектуальных агентов и мультиагентных систем представляют собой одну из наиболее интересных областей развития современных информационных технологий. По оценкам специалистов в области искусственного интеллекта агентно-ориентированный подход может сыграть важнейшую роль в становлении информационного общества будущего. Уже сегодня он широко применяется в таких областях как управление информационными потоками (workflow management), диагностика и управление в сетевых структурах (network management), информационный поиск (information retrieval), электронная коммерция (e-commerce), обучение (education) и ряде других.

Существует большое количество определений понятия "программный агент" в зависимости от взгляда на распределенную обработку знаний. С точки зрения распределенных вычислений, агент – это самостоятельный процесс, имеющий определенное состояние и способный взаимодействовать с другими агентами посредством передачи сообщений. В этом смысле его можно рассматривать как естественное развитие парадигмы объектно-ориентированного параллельного программирования. По существу программные агенты выполняют задачи, делегируемые им пользователями. В зависимости от поставленной задачи, агенты могут размещаться на Web-серверах, настольных персональных компьютерах или в локальных сетях.

Другая точка зрения ассоциируется с областью распределенного искусственного интеллекта, где термин "агент" имеет более узкое и специфическое значение, чем упомянутое выше. В соответствии с ним, "агент" – это компьютерная система, которая в дополнение к перечисленным выше свойствам реализована с использованием концепций, свойственных человеку. Например, с такими понятиями, как знания, убеждения, намерения, обязательства [1].

Цель исследования

Целью проводимых исследований является разработка математической модели программного агента, которая позволит реализовать многоуровневую, иерархическую систему автономного администрирования распределенных информационных ресурсов вычислительной системы.

Рассмотрим один из вариантов построения модели управления информационными ресурсами на основе мультиагентной системы. Главными составляющими такой системы являются:

– агент – приложение, функционирующее на выделенном ему компьютере и используемое для сбора, анализа, защиты информации;

– система управления агентами (менеджер агентов);
– программа-администратор мультиагентного пространства, осуществляющая контроль работы системы, накопления и анализа информации, полученной от агентов в процессе их функционирования;

– конструктор агентов – программный модуль, обеспечивающий логическое и физическое конструирование агентов. Он отвечает за построение, генерацию и запуск агентов на удаленные компьютеры;

– библиотека функций – набор модулей, из которых строится программный агент. Каждая конкретная логическая функция на этапе проектировании должна иметь реализацию в виде программного модуля;

– задание – запрограммированное одно или несколько действий (сценарий), которые должен выполнить программный агент по команде пользователя или в результате работы собственной экспертной системы [2].

Разработка модели программного агента

В качестве математической модели программного агента может быть рассмотрена модель в виде фреймовой структуры. В общем случае такую модель можно представить в виде:

$$FR\{< R_1, A_1 >,... < R_2, A_2 >, < R_k, A_k >\}, \quad (1)$$

где FR – имя фрейма (агента); пара $<R_i,A_i>$ – i-й слот фрейма; R_i – имя слота, A_i – значение слота.

Значение слота может быть представлено последовательностью:

$$\{< W_1, D_1 >,... < W_n, D_n >, < S_1 >; < S_m >\}. \quad (2)$$

Здесь Wi – имена атрибутов слота; Di – значение атрибутов слота; Si – ссылки на другие слоты.

Привлекательность модели (1)–(2) состоит в следующем. Фрейм в предлагаемом подходе выступает в виде универсального каркаса или типовой оболочки, в которую могут добавляться функциональные модули-слоты для решения конкретных задач информационной поддержки.

Рассмотрим пример логической модели программного агента с именем AG, который выполняет две задачи и, соответственно, состоит из двух слотов.

Каждый слот R1, R2 имеет одинаковую структуру и содержит три атрибута. Первый атрибут слота R1 определяет объект действия – файл с именем polis.doc. Второй атрибут определяет условие, при котором должно состояться выполнение операции, если размер файла достигнет 50 Кбайт.

Третий атрибут определяет вид операции или вид действия – копировать файл polis.doc на диск D:\ в папку MAIN. Слот R2 выполняет задачу регистрации доступа (время изменения данных) к файлу базы данных c:\BASE\STUD.mdb, протокол доступа хранит в файле d:\MAIN\hist.doc.

Логическая структура программного агента с именем AG может быть представлена в виде:

Имя слота	A1: объект действия	A2: условие	A3: тип действия
R1	C:\TEMP\POLIS.doc	IF size > 50kb	COPY to D:\MAIN
R2	C:\BASE\STUD.mdb	–	PROT D:\MAIN\hist.doc

Предложенный подход основан на формализациях, используемых в теории образов [3].

Введем основные понятия в предлагаемом подходе:

– компонент-слот программного агента;

– функциональная база компонентов-слотов;

– класс программных агентов;

– структура семейства программных агентов.

Рассмотрим математическую трактовку основных понятий.

Задачи, решаемые программным агентом, определяют его структуру и минимально необходимое число слотов – составных частей, обеспечивающих его функционирование. Унифицированные составные части агента – слоты образуют функциональную базу создаваемого семейства E программных агентов. Функциональная база делится на непересекающиеся классы E_i компонентов-слотов, выполняющие различные функции информационной поддержки вычислительной системы.

Компонент-слот e функциональной базы образует в программных агентах (фреймах) связи с другими компонентами-слотами. Для осуществления этих связей он может быть снабжен программными элементами сопряжения. Структуру компонента-слота можно представить графом, содержащим набор изолированных вершин, каждая из которых соответствует элементу сопряжения.

Любому компоненту-слоту e соответствует определенная арность $Dim(e)$, которая показывает число модулей сопряжения (число изолированных вершин графа компонента). Модули сопряжения компонентов-слотов могут быть описаны набором признаков или параметров.

Таким образом, одним из факторов, определяющих функциональные возможности компонентов-слотов каждого класса, является наличие и типы встроенных модулей сопряжения. Другой фактор – это различная программная реализация компонентов-слотов и тип потребляемого в режиме функционирования ресурса. Конкретной программной реализации компонента-слота e функциональной элементной базы соответствует его представление вектором параметров $V(e)$.

Любой компонент-слот e функциональной базы программных агентов можно представить следующими образом:

$$e = [I(e), G(e), V(e)], \quad (3)$$

где $I(e)$ – номер класса компонента-слота; $G(e)$ – тип структуры модулей сопряжения; $V(e)$ – вектор конструктивных параметров.

Для формального определения возможных функциональных вариантов компонентов-слотов введем преобразование подобия s, отображающее множество E на себя $s : E \to E$. Два образующих компонента-слота e_1 и e_2 подобны, если существует некоторое преобразование подобия $s \in S$, такое, что $se_1 = e_2$. Множество S содержит среди прочих преобразование подобия вида s_0, для которого $s_0 e = e$. Конкретный вид преобразований подобия зависит от проектируемого семейства программных агентов.

Будем считать, что используемые преобразования подобия удовлетворяют следующим требованиям:

– множество S преобразований подобия s является группой;

– любое $s \in S$ отображает класс E_i в себя при любом номере класса компонентов-слотов;

– элементы множества S не влияют на структуру модулей сопряжения, но могут влиять на их признаки.

Таким образом, структура модулей сопряжения и индекс класса компонентов инвариантны относительно преобразования подобия.

С учетом свойства инвариантности для компонента-слота можно ввести понятие его функционального класса: $z = [I(e), G(e)]$, которое отражает наиболее важную информацию о компонентах-слотах, заключенную в номере класса и структуре модулей сопряжения. Для функциональной базы E можно определить интегральную категорию верхнего уровня – функциональную базу типов.

Рассмотрим отображение $h : E \to E'$, где E и E' – две функциональные базы одного и того же семейства, связанные с одной и той же группой преобразований подобия S.

Будем называть преобразование h инвариантом связей, если для любого компонента-слота $e \in E$ компоненты e и $e' = h(e)$ имеют одни и те же программные модули сопряжения, а также $se \to se'$ для $\forall s \in S, \forall e \in E$.

С помощью отображения h можно осуществить переход от функциональной базы E к новой элементной базе E', не нарушая общей структуры программных агентов, в которых используются компоненты E.

Данный случай возникает при замене имеющейся функциональной базы семейства элементной базой нового поколения. Структура элементной базы семейства формируется классами компонентов-слотов. Внутри класса слоты отличаются структурой модулей сопряжения $G(e)$. Компоненты одного класса, отличающиеся какими-либо из перечисленных выше характеристик и признаков, будем называть модификациями компонентов-слотов класса. Модификации компонентов учитываем различными их номерами; так, при двухсимвольном обозначении e_{ij} первый символ указывает класс компонента-слота, второй – номер модификации.

Компоненты класса, связанные с другими его компонентами одним и тем же преобразованием подобия S, являются универсальными. Число универсальных компонентов-слотов в классе определяется числом используемых преобразований подобия.

Программные агенты (фреймы) формируются в зависимости от типа поставленной администратором системы задачи путем подбора и установки компонентов-слотов функциональной базы E. Они отличаются друг от друга составом компонентов-слотов и структурой фрейма-прототипа $FR = Q(e_1, e_2 ... e_n)$. Структура программного агента – это множество соединений, существующих между базовым фреймом-прототипом и узлами сопряжения входящих в него компонентов-слотов. Фрагмент построения программного агента представлен на рис.1.

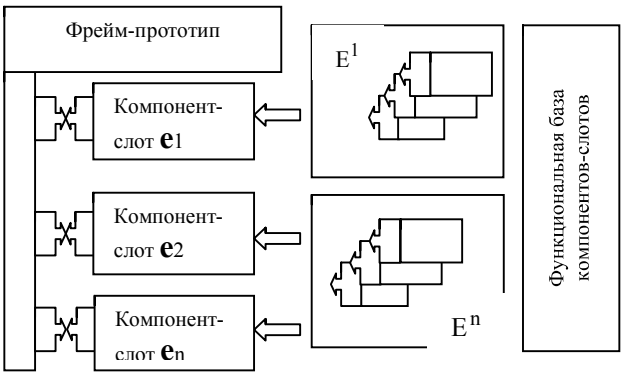

Рис. 1. Структура программного агента

Два программных агента идентичны, если у них совпадают составы компонентов-слотов и структуры модулей сопряжения.

Множество структур программных агентов Q, получаемых из компонентов-слотов функциональной базы, будем называть Q – классом. Класс программных агентов представляет собой множество графов, описывающих модули сопряжения компонентов-слотов, связанных друг с другом [4].

Для иллюстрации предлагаемого подхода рассмотрим технологию доступа к распределенным ресурсам информационной системы на основе агентной технологии. Для решения поставленной задачи на каждом локальном персональном компьютере сети поместим программного агента, состоящего из функциональных компонентов-слотов. При этом один программный агент может решать несколько задач по управлению доступом к ресурсам данного персонального компьютера.

Фрагмент такой системы представлен на рис.2.

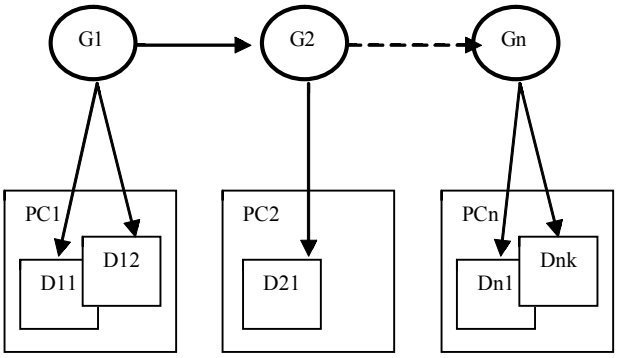

Рис. 2. Мультиагентная система автономного администрирования информационных ресурсов: PC1 – PCn – персональные компьютеры локальной вычислительной системы; Dij – информационные ресурсы i-го компьютера; Gi – программный агент, выполняющий роль посредника для организации доступа к ресурсам i-го компьютера

Выводы

В результате проведенных исследований предложена модель программного агента на основе фреймовой структуры. Слоты в рассмотренной модели выполняют роль функциональных модулей, решающих конкретные задачи информационной поддержки. Обоснован формальный подход к задаче проектирования логической структуры взаимосвязанных программных агентов. Предложенные модели могут использоваться для разработки мультиагентных систем автономного администрирования информационных ресурсов распределенных вычислительных систем.

Литература: 1. *Foundation* for Intelligent Physical Agents (FIPA) Spec: DRAFT, version 0.2, Agent Communication language, 1999. 322 p. **2.** *Пономаренко Л.А., Філатов В.О.* Програмні агентні технології в адмініструванні баз даних // Вісник Київського торговельно-економічного університету. Київ. Вип.3/2001. С.68-73. **3.** *Гренавдер У.* Лекции по теории образов: Пер. с анг. / Под ред. Ю.И. Журавлева. М.: Мир, 1979. 256с. **4.** *Конноли Т., Бегг К.* Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер с англ. М.: Вильямс, 2000. 1120с.

Поступила в редколлегию 09.02.2003

Рецензент: д-р физ.-мат. наук, проф. Яковлев С.В.

Филатов Валентин Александрович, канд. техн. наук, доцент кафедры искусственного интеллекта ХНУРЭ. Научные интересы: распределенные информационные системы, базы данных, агентные технологии. Увлечения: автоспорт, футбол. Адрес: Украина, 61031, Харьков, ул. Ромашкина, 6/а, кв. 19, e-mail: Filatov_val@ukr.net

УДК 681.5

СИНТЕЗ ИНВАРИАНТНЫХ КЛАССИФИКАТОРОВ

ПОЛОНСКИЙ А.Д.

Для идентификации случайных аналоговых сигналов в условиях статистической неопределенности функционирования измерительных каналов систем управления описывается метод синтеза инвариантных классификаторов на основе обучения распознавания медианных отношений между наблюдениями. Предлагается ряд схем инвариантных классификаторов, синтезированных в элементном базисе рангеров.

Введение

В системах управления применяются специализированные процессоры, которые называются классификаторами [1]. На основании выводов классификатора регулятор принимает управленческое решение. Качество такого управления существенно зависит от эффективности решения задачи синтеза классификатора. В основу решения этой задачи положены методы обучения распознавания образов [1, 2]. На практике, когда в измерительных каналах систем управления действуют случайные аналоговые сигналы, возникает проблема обучения распознавания образов в условиях априорной неопределенности. Суть этой проблемы состоит в том, что для классических методов обучения распознавания образов [2] нет, как правило, полного объема априорных сведений о статистических свойствах случайных сигналов. В связи с этим возникает задача синтеза классификаторов, которые обладают свойством инвариантности к распределениям случайных аналоговых сигналов. В настоящей работе для синтеза инвариантных классификаторов предложен метод обучения распознавания отношений. В элементом базисе рангеров синтезированы электрические схемы инвариантных классификаторов.

1. Постановка задачи

Будем рассматривать классификаторы как идентифицирующие системы вида

$$z_i = a_i y_i = \begin{cases} 0 \mid U \notin R_i \notin Y; \\ y_i \mid U \in R_i \notin Y, \end{cases} \quad i = \overline{1, m}. \qquad (1)$$

Здесь z_i есть многоместная (n-арная) функция, заданная на множестве идентифицирующих переменных U={u(1), ..., u(n)} со значениями из множества $\{0, y_i\}$; $Y = \{y_1, K, y_m\}$ – множество идентифицируемых переменных; $R_i \in U^n$ – заданное на множестве U отношение, для которого $a_i = 1$, $a_j = 0\ \forall j \neq i = \overline{1, m}$; U^n – n-я декартова степень множества U; a_i – весовые коэффициенты, подлежащие определению.

В дальнейшем изложении под идентифицирующими переменными понимаются независимые мгновенные значения напряжений случайных аналоговых сигналов u(1), ..., u(n) с произвольными интегральными функциями распределений (ИФР) вида $F(U) = \prod_{i=1}^{n} F(u(i))$.

Здесь F(U) и F(u) есть соответственно n-мерные и одномерные ИФР мгновенных значений напряжений случайных аналоговых сигналов.

Выражения (1) при $R_1 \cup R_2 \cup K \cup R_m = U^n$ определяют класс алгоритмов идентификации одной y_i из m переменных y_1, K, y_m по признаку, заданному отношением

$$R_i = \{u(i_1) o u(i_2) o K\ o u(i_{n-1}) o u(i_n)\}. \qquad (2)$$

Здесь $u(i_p) o u(i_q)$ есть либо $u(i_p) > u(i_q)$, либо $(i_p) < u(i_q)$; $\{i_1, K, i_n\}$ – i-я перестановка целых чисел от 1 до n.

Задача синтеза рассматриваемых классификаторов состоит в том, что требуется определить весовые коэффициенты, которые соответствуют заданным отношениям между идентифицирующими переменными.