

МОДЕЛЬ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА КАК АБСТРАКТНЫЙ ТИП ДАННЫХ

У статті запропоновано підхід до побудови інтелектуального агента як абстрактного типу даних. Розглянутий підхід базується на теорії алгебраїчних систем. Ієрархічна модель програмного агента як абстрактного типу даних має в основі концепцію типів даних. Логічна модель інтелектуального агента запроваджена на підставі фрейму. Запропонований підхід дозволяє інтелектуальному агенту ефективно взаємодіяти з типами та структурами даних сучасних інформаційних систем.

Введение

Современные достижения в области информационных технологий позволяют решать задачи управления ресурсами информационных систем и распределенных баз данных. Реализовать комплекс задач администрирования можно с помощью особого класса управляющих систем, функционирующих на основе технологии workflow. Поток работ workflow – это операции, включающие координируемое выполнение множества задач, выполняемых различными обрабатываемыми объектами. Задача определяет некоторую работу, которую необходимо выполнить, и она может быть специфицирована различными способами, включая текстовое описание в сообщении по электронной почте или компьютерную программу. Выполнение множества задач с привлечением различных обрабатываемых объектов осуществляет специализированный программный комплекс – система управления потоками работ.

Системы управления потоками работ на основе технологии программных агентов и мультиагентных систем (СУПР МАС) являются сравнительно новой категорией. Согласно принципам функционирования таких систем управление информационными ресурсами осуществляется множеством автономных интеллектуальных агентов, которые кооперируются для достижения поставленной цели. Эффективность функционирования СУПР МАС во многом зависит от выбора типа модели программного агента, совокупности средств определения допустимых структур агента и множества операций с данными информационной среды. Схема взаимодействия программного агента с информационной средой представлена на рис. 1.

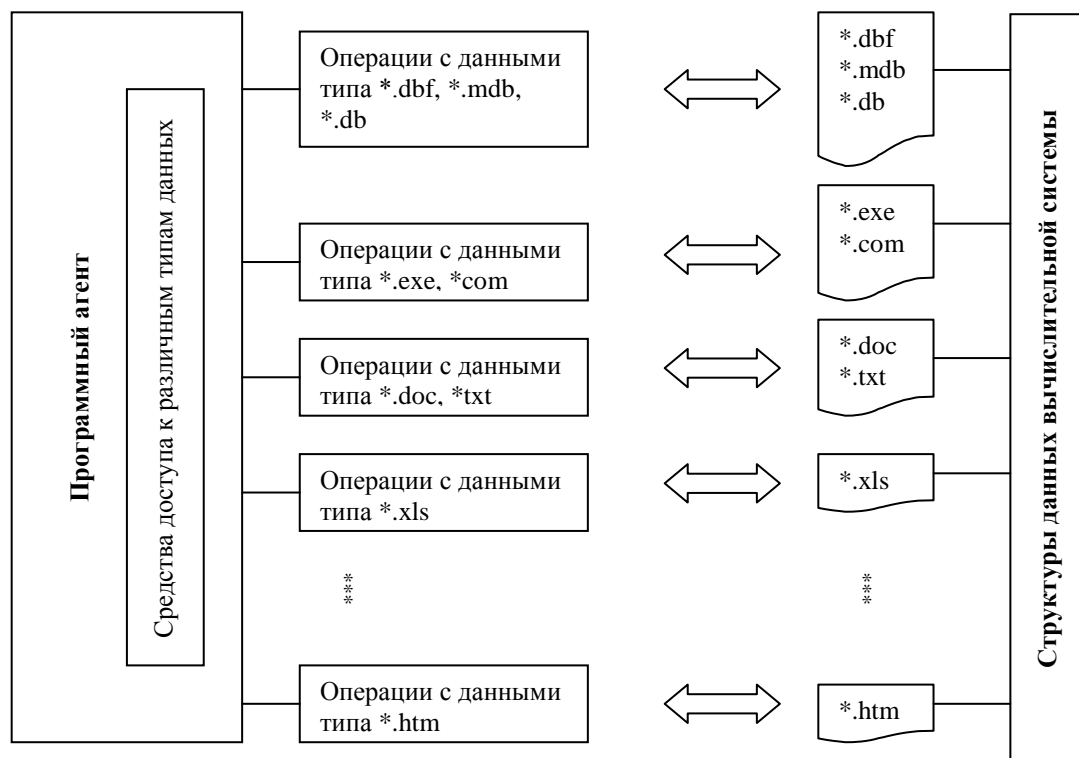


Рис. 1 Схема взаимодействия программного агента с информационной средой

Цель исследования

Целью проводимых исследований является разработка формализованного подхода к проектированию систем управления потоками работ на основе мультиагентных технологий.

Мультиагентной системой управления потоками работ будем называть многокомпонентную систему, состоящую из программных агентов, выполняющих локальные задачи информационной поддержки для достижения общей цели – управление распределенной информацией.

Основные составляющие мультиагентной системы:

Агент – приложение, функционирующее на локальном компьютере и используемое для реализации взаимосвязи с информационным ресурсом.

Система управления агентами (менеджер агентов) – программа-администратор мультиагентного пространства, осуществляющая следующие функции:

- контроль работы системы;
- дистанционное управление поведением мультиагентной системой;
- модификация и настройка функций программных агентов на удаленных персональных компьютерах;
- накопление и анализ информации, полученной от агентов в процессе их функционирования.

Конструктор агентов – подсистема, обеспечивающая логическое и физическое проектирование агентов, их генерацию, модификацию и запуск на удаленные компьютеры.

Разработка и исследование модели агента как типа данных

Рассмотрим программного агента как некоторую абстрактную алгебраическую систему. В общем виде ее можно представить как систему:

$$U = \langle A, \Omega_f, \Omega_p \rangle, \quad (1)$$

где A – множество основ; $\Omega_f = \langle F_1, \dots, F_k \rangle$ – множество операций, заданных на множестве A ; $\Omega_p = \langle P_1, \dots, P_m \rangle$ – множество предикатов, заданных на множестве A . Система U может быть записана в кратком виде $U = \langle A, \Omega \rangle$, при условии объединения множеств $\Omega = \Omega_f \cup \Omega_p$.

В рассмотренной постановке (1), концепция программного агента должна предполагать наличие универсального языка описания данных и средств манипулирования на всем диапазоне типов данных информационной системы от текстовых файлов до файлов баз данных. Такой иерархический подход к построению модели программного агента является нерациональным по следующим соображениям. Основные типы файлов и их структуры (*.txt, *.xls и т.д.), как правило, разрабатывались под специализированные программные системы: так файлы типа *.doc для тестового редактора WORD, файлы типа *.dbf – для систем управления базами данных. Эти приложения, за редким исключением, сами представляют собой замкнутые алгебраические системы.

В последнее время внимание исследователей было сосредоточено на выработке понятия типа данных с учетом принципа иерархического построения, не привязываясь жестко к инициальным или терминальным алгебрам. Иерархический тип данных конструируется на основе одного или нескольких примитивных типов (которые, в свою очередь, могут быть иерархическими), входящих в исходную систему типов данных. Каждый примитивный тип может быть проанализирован и реализован отдельно, без использования информации о тех типах, для построения которых он будет использован. С другой стороны, и вновь создаваемый тип может рассматриваться как «черный ящик», поведение которого оценивается в терминах примитивных типов.

Исходя из изложенного, целесообразно рассмотреть концепцию модели программного агента, как алгебраическую систему, состоящую из непересекающихся замкнутых подсистем.

Утверждение. В алгебраической системе $U = \langle A, \Omega \rangle$ непустое множество $A_1 \in A$ называется замкнутым, если A_1 замкнуто относительно каждой операции F_k этой системы.

Это означает, что результат любой операции, произведенной над произвольными элементами множества A_1 , принадлежит A_1 .

Обозначим через F'_k и P'_m соответственно операции и предикаты, определенные на A_1 , значения которых совпадают со значениями операций $F_k \in \Omega$ и предикатов $P_m \in \Omega$.

В результате получим алгебраическую систему $U_1 = \langle A_1, \Omega' \rangle$, являющуюся подсистемой или подмоделью системы U . Подсистема U_1 однозначно определяется множеством A_1 и, следовательно, вместо $U_1 = \langle A_1, \Omega' \rangle$ можно употреблять $U_1 = \langle A_1, \Omega \rangle$.

Рассмотренный выше подход к построению модели программного агента позволяет представить ее как абстрактный тип данных в терминах непересекающихся подтипов.

Для эффективного решения задачи управления информационными ресурсами распределенной системы целесообразно файлы, приложения и другие компоненты управляемой системы представлять в виде управляемых объектов. Объект характеризуется множеством текущих состояний – X_i , возмущающих и управляющих воздействий – U_i, Z_i выходных переменных Y_i . При этом $Y_i = F_i(X_i, U_i, Z_i)$ $i = \overline{1, N}$, где N – количество управляемых информационных объектов.

Примером управляемого объекта может служить произвольный файл стандарта WORD. Множество его состояний:

- $X = \langle \text{размер файла, дата создания, адрес, автор} \rangle$;
- несанкционированные, случайные действия пользователей – $Z = \langle \text{дата изменения, характер изменения, размер, пользователи, уровень доступа} \rangle$;
- допустимые операции с объектом $U = \langle \text{дата изменения, размер, уровни доступа, список пользователей} \rangle$;
- $Y = \langle \text{новый размер, новая дата изменения, новый пользователь, новый адрес} \rangle$.

Предложенный подход представления ресурсов информационной системы как управляемого объекта позволяет эффективно осуществлять его взаимодействие с программным агентом.

Рассмотрим модель программного агента, представленную в виде фреймовой структуры. В общем случае такая модель может быть записана в виде:

$$FR \langle R_1, A_1 \rangle, \dots \langle R_2, A_2 \rangle, \langle R_k, A_k \rangle, \quad (2)$$

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

На основании фрейма (2), может быть реализована модель программного агента в терминах $\langle \text{объекты} \rangle$, $\langle \text{условия} \rangle$, $\langle \text{действия} \rangle$, $\langle \text{приоритет} \rangle$. Фрейм выступает в виде универсального каркаса или типовой оболочки, в которую могут добавляться функциональные модули-слоты для решения конкретных задач администрирования информационных ресурсов. Каждый слот может формироваться из четырех атрибутов базовых типов и выполнять требуемые операции с данными.

Логическая модель программного агента представлена в табл.1 Слот A_1 содержит четыре атрибута. Первый атрибут определяет объект действия – файл с именем `res.doc`. Второй атрибут определяет условие, при котором должно состояться выполнение операции, – если размер файла достигнет 90 Килобайт. Третий атрибут определяет вид операции или вид действия – копировать файл `polis.doc` на диск `D:\` в папку `IN`. Четвертый атрибут определяет приоритет – важно. Слот A_2 выполняет задачу регистрации доступа (время открытия) к файлу базы данных `c:\BASE\ST.mdb`, протокол доступа хранит в файле `d:\Cont\pro.doc`. Приоритет – конфиденциально.

Таблица 1.

A1	OBG: c:\MP\res.doc	CON:IF size > 90kb	ACT: COPY to d:\IN	STA: IMP
A2	OBG: c:\BASE\ST.mdb		ACT:PROT d:\Cont\pro.doc	STA: CONF

Выводы

В статье рассмотрены вопросы построения модели программного агента для автономного администрирования информационных ресурсов как абстрактного типа данных. Теоретически обоснован подход на основании концепции алгебраических систем. Предложенная модель интеллектуального агента модели может использоваться для разработки мультиагентных систем и позволяет эффективно решать комплекс задач мониторинга и защиты информационных ресурсов вычислительной системы.

ЛИТЕРАТУРА:

1. Системы программирования баз данных и знаний / Замулин А.В. – Новосибирск: Наука, 1990. – 352с.
2. Мальцев А.И. Алгебраические системы. – М.:Наука, 1970. – 392с.
3. Пономаренко Л.А., Филатов В.А., Цыбульник Е.Е. Динамическое администрирование баз данных с использованием агентных технологий – Научный сборник "Труды Одесского политехнического университета". Вып.4 (16), Одесса –2001. – с.95-97.
4. Пономаренко Л.А., Филатов В.О. Програмні агентні технології в адмініструванні баз даних. – Вісник Київського торговельно-економічного університету. - Київ.- вип.3/2001, с.68-73.

МОДЕЛЬ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА ЯК АБСТРАКТНИЙ ТИП ДАНИХ

В.А. Філатов, Л.Е. Чалая

У статті запропоновано підхід до побудови інтелектуального агента як абстрактного типу даних. Розглянутий підхід базується на теорії алгебраїчних систем. Ієрархічна модель програмного агента як абстрактного типу даних має в основі концепцію типів даних. Логічна модель інтелектуального агента

запроваджена на підставі фрейму. Запропонований підхід дозволяє інтелектуальному агенту ефективно взаємодіяти з типами та структурами даних сучасних інформаційних систем.

INTELLECTUAL AGENT'S MODEL AS AN ABSTRACT TYPE OF THE DATA

V. Filatov, L. Chalaya

In this article the approach to construction the intellectual agent's model as an abstract data's type is offered. The described approach is based on the theory of algebraic systems. The concept of the data's types is fixed in a basis of hierarchical model of the program agent. The frame is fixed in a basis of the intellectual agent's logic model. The given approach allows the intellectual agent effectively to cooperate with the data's types and structures of the modern information systems.

МОДЕЛЬ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА КАК АБСТРАКТНЫЙ ТИП ДАННЫХ

В.А. Филатов, Л.Э. Чалая

В статье предложен подход к построению модели интеллектуального агента как абстрактного типа данных. Рассмотренный подход основан на теории алгебраических систем. Иерархическая модель программного агента как абстрактного типа данных разработана на основе концепции типов данных. Логическая модель интеллектуального агента имеет фреймовую структуру. Предложенный подход позволяет интеллектуальному агенту эффективно взаимодействовать с типами и структурами данных современных информационных систем.