

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

Предложен подход для выбора агентов интеллектуального интерфейса информационных систем, основанный на установлении соответствия функций и требований к информационным системам, к агентам и к инструментальным средствам для их разработки. Примеры структурирования информации на различных этапах выбора представлены с ориентацией на автоматизированную систему проектирования.

© Ю.С. Яковлев, Л.И. Курзанцева,
И.М. Першко, 2005

УДК 681.324

Ю.С. ЯКОВЛЕВ, Л.И. КУРЗАНЦЕВА, И.М.
ПЕРШКО

О ВЫБОРЕ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ ИНТЕРФЕЙСОВ ИНФОРМАЦИОННЫХ СИСТЕМ

Введение. Одной из главных задач при создании информационных систем (ИС) является разработка социально ориентированного на определенные группы пользователей интеллектуального интерфейса (ИИ). С этой точки зрения важное значение приобретает структурирование и анализ исходной информации, которая освещала бы наиболее распространенные средства и способы построения интеллектуального интерфейса информационных систем и давала бы возможность разработчику в определенной степени сориентироваться в особенностях подхода к его построению. Учитывая, что одними из перспективных ИС являются многоагентные системы (МАС) и соответственно многоагентные технологии построения интеллектуального интерфейса, следует обратить внимание разработчиков на проблемы, имеющие место при создании и применении ИИ для МАС, и высказать некоторые соображения по выбору соответствующих языков программирования и инструментальных программных средств для создания агентов в рамках информационных систем с искусственным интеллектом.

Особенности реализации ИИ для различных типов интеллектуальных информационных систем (ИИС).

Функции и средства поддержки интеллектуального интерфейса и особенности его реализации прежде всего определяются типом и назначением ИИС, разновидности которых (с учетом [1]) показаны на рис. 1.

Естественно-языковой интерфейс транслирует естественно-языковые конструкции на



РИС. 1. Разновидности ИИС, использующие интеллектуальный интерфейс

уровень внутреннего представления информации и обратно. Такой интерфейс применяется для доступа к интеллектуальным базам данных, контекстного поиска текстовой информации, голосового ввода команд в системах управления, машинного перевода с иностранных языков.

Гипертекстовые системы реализуют для пользователя поиск в базах данных текстовой и мультимедийной информации по ключевым словам.

Системы контекстной помощи можно рассматривать как своего рода объединение интеллектуальных гипертекстовых и естественно-языковых систем. В отличие от обычных систем помощи, предлагающих пользователю схему поиска требуемой информации, с помощью диалога с пользователем они уточняют проблему (ситуацию) и осуществляют поиск рекомендаций, относящихся к конкретной ситуации.

В системах когнитивной графики интерфейс реализуется с помощью графических образов, генерируемых в соответствии с происходящими событиями.

Из всего многообразия систем, показанных на рис. 1, акцентируем внимание на многоагентных системах (как наиболее перспективных) и соответственно на интерфейсных агентах, которые можно классифицировать в зависимости от различных признаков, например, в зависимости от выполняемых ими функций, используемой технологии, среды обитания и т.д. Один из вариантов классификации интерфейсных агентов в зависимости от выполняемых ими функций показан на рис. 2 [2].

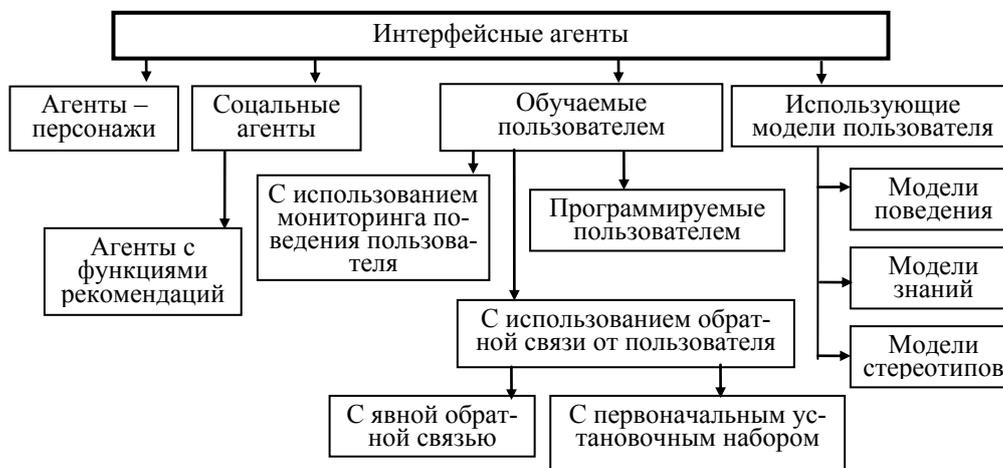


РИС. 2. Вариант классификации интерфейсных агентов

Современные интеллектуальные системы, как правило, строятся таким образом, что они обеспечивают поддержку одновременно нескольких разновидностей интерфейсных агентов с различными функциями. В табл. 1 приведены примеры систем, в которых интерфейсные агенты (согласно назначению систем) реализуют определенные наборы функций, показанные на рис. 2 [2]. В частности, система Act, применяемая в индустрии развлечений, ведет наблюдение за пользователем и оказывает ему помощь в выполнении новых действий по образцу старых. Система Maxims осуществляет фильтрацию электронной почты на основе изучения действий пользователя. Coach контролирует действия пользователя и при выявлении ошибочных действий предлагает советы. Meeting scheduling agent занимается планированием встреч на основании повторяющихся действий пользователя. Система New T фильтрует статьи из группы новостей с применением модели векторного пространства. Fab-система с функциями рекомендаций производит поиск Web-страниц на основе получаемой обратной связи согласно контекстному анализу полученных документов. Система Let's Browse оказывает помощь пользователям TV- систем в просмотре Web, создавая начальный профиль пользователя. Система CILA – поддерживает контроль за действиями пользователя, обладает способностью к самообучению и сотрудничает с другими агентами.

Анализируя информацию подобного рода, можно построить таблицу соответствия функций интеллектуального интерфейса (ИИ), необходимых базовых функций используемых для этого интерфейсных агентов (ИА) и способа реализации этих функций. Один из возможных вариантов такого соответствия приведен в табл. 2.

При создании программных агентов, реализующих заданные функции, необходимо выбрать язык программирования агента, язык межагентного взаимо-

действия, а также технологии, поддерживающие среду обитания агентов, и инструментальные средства для их разработки.

ТАБЛИЦА 1. Разновидности интерфейсных агентов, используемых конкретными системами

Интерфейсные агенты с соответствующими функциями (согласно рис. 2)	Примеры систем						
	ACT	Maxims	Fab	CILA	Coach	Let's Browse	New T
Агенты –персонажи	+						
Социальные агенты		+					
Агенты с функциями рекомендации			+				
Агенты, обучаемые пользователем с использованием для обучения							
мониторинга поведения пользователя	+	+	+	+	+	+	
обратной связи			+	+			+
установочного набора				+		+	+
программируемые пользователем		+					+
Агенты, использующие модели пользователя с применением							
поведенческой модели	+	+		+		+	+
модели знаний					+		
моделей стереотипов					+		

Для программирования агентов могут применяться: универсальные языки (Java, C++ , Visual Basic и др.), языки представления знаний (SL, KIF), языки переговоров и обмена знаниями (KQML, AgentSpeak, April), языки сценариев (Tcl/Tk, Python, Perl 5 и др), специализированные языки (TeleScript, COOL, Agent0, AgentK и др.), символьные языки и языки логического программирования (Oz, ConGolog, IMPACT, Dylog, Concurrent METATEM, Ehhf и др.), а также другие языки и средства разработки агентов, основные характеристики которых приведены в табл. 3, построенной с учетом данных [3 – 12].

В табл. 4 приведено сравнение параметров отдельных инструментальных средств, обозначенных в табл. 3, которые используются для разработки агентов [13].

Анализ табл. 3 и 4 показал, что средства для разработки агентов можно условно разделить на две группы: средства, построенные на базе языка Java, и средства, построенные на базе других языков. Средства первой группы предназначены для разработки сетевых приложений на базе мобильных агентов, взаимодействующих через протокол TCP/IP.

ТАБЛИЦА 2. Пример построения таблицы соответствия функций ИИ и функций интерфейсных агентов, используемых для их реализации

Функции ИИ	Функции ИА	Способ реализации функций ИА
Коммуникация	Общения с пользователем на естественном языке	Использование невербальной модели (восприятие голоса), языковой модели (распознавание слов). Распознавание голосовых сообщений, жестов пользователя
Масштабируемость функций	Инициализация и самоуничтожение агента	Формирование программного агента и его уничтожение по завершению задания
Ввод, обработка и выдача информации пользователю	Автоматизация процесса продвижения информации и выдача рекомендаций	Прием, обработка и ранжирование информации, принятие решения о дальнейшем её продвижении. Анализ состояния ИИС, выдача рекомендации пользователю путем логических выводов
Самообучение и предоставление помощи	Мониторинг действий пользователя	Анализ действий пользователя, прогноз возможных последствий действий, корректировка и блокировка работы пользователя
	Оптимизация информационной загрузки пользователя	Контроль занятости пользователя, его зоны внимания. Фильтрация и распределение информационных потоков: электронной и голосовой почты, интернет-новостей и др.
Адаптивность	Автоматическая адаптация к условиям и потребностям пользователя	Создание модели пользователя, настройка внешнего вида интерфейса и алгоритмов работы системы под потребности пользователя
	Моделирование эмоциональных состояний личности	Определение состояния пользователя и изменение способа общения с ним в зависимости от уровня усталости, распознавание уровней агрессии

ТАБЛИЦА 3. Примеры средств разработки агентов

Название языка (средства разработки), URL-адрес	Краткая характеристика / назначение
1	2
AgentTalk http://www.kecl.ntt.co.jp/csl/msrg/topics/at	Средство для разработки автономных мобильных агентов на базе языка сценария Tcl с поддержкой Tcl-инструментария для создания графического интерфейса
AgentTool en.afit.af.mil/ai/agentool.htm	Агентная структура на базе Java. Использование GUI для конструирования системы
:	:
:	:

Окончание табл. 3

1	2
Cybele http://www.opencybele.org	Средство для разработки распределенных агентных приложений на базе Objective C, с поддержкой для C и C++
FIPA-OS fipa-os.sourceforge.net www.nortelnetworks.com/products/announcements/fipa/	Реализация элементов, содержащихся в FIPA-спецификации по взаимодействию агентов. Java-реализация архитектуры для разработки агентов
Echelon	Средство для разработки агентов в сети LonWorks.
JAFMAS http://www.ececs.uc.edu/~abaker/JAFMAS/	Программа, обеспечивающая создание мультиагентной системы на Java. Поддерживает межагентное общение на KQML
Remembrance Agents rhodes.www.media.mit.edu/people/rhodes/RA/	Средство для разработки агентов, наблюдающих за пользователем и предлагающих информацию, необходимую в данный момент
⋮	⋮
Ummon http://www.spacetime.com/projects/ummon/	Средство для создания самообучающегося агента. Содержит методы ИИ для достижения "человеческого" общения
Virtual Secretary Project (ViSe) http://www.cs.uit.no/DOS/Virt_Sec/	Создание модели пользователя на основе интеллектуальных агентов для выполнения секретарских задач (технология Tcl/TclX/Tix/Tk)

ТАБЛИЦА 4. Сравнение параметров инструментальных средств разработки агентов

Параметр	Наименование инструментальных средств					
	JAFMAS	AgenTalk	AgentTcl	Telescript	Swarm	Echelon
Язык разработки	Java	Lisp	Tcl	Telescript	Objective C	Silicon chip
Поддержка ОС	Все	UNIX	UNIX	UNIX	UNIX	Echlon chip
Реализация агента	Пр.	Пр.	Пр.	Пр.	Пр.	Ап.
ООП	+	-	-	+	+	+
Наличие у агента своего плана действий	+	-	+	+	+	+
Коммуникационный протокол	TCP/IP и UDP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP
Мобильность агентов	+	-	-	-	-	-

Примечание: Пр. – программная; Ап. – аппаратная; ООП – объектно-ориентированное программирование.

Благодаря тому, что Java архитектурно-независимый язык, компилирующий свой код в машинно-независимый, приложения, разработанные с помощью средств этой группы, могут работать на многих других платформах. Различие между средствами в том, что не все предоставляют поддержку речевого обмена сообщениями с помощью KQML. Недостатком является отсутствие классов для определения социального поведения агентов.

Инструментальные средства второй группы, в основном, предназначены для проектирования сложных динамических агентных структур и реализации сред, хотя имеются отдельные экземпляры для создания мобильных приложений (TeleScript, Agent Tcl). При этом агенты разрабатываются согласно BDI-модели. Коммуникации осуществляются через протокол TCP/IP. Однако такие средства имеют слабые возможности по согласованию и сотрудничеству между агентами и работают на ограниченном количестве платформ, поскольку язык программирования – не унифицированный.

Большинство коммуникативных агентных языков (Agent Communication Language – ACL) основаны на речевом взаимодействии (речевые действия выражаются посредством стандартных ключевых слов) [14]. На рис. 3 показана схема варианта классификации языков межагентных коммуникаций.

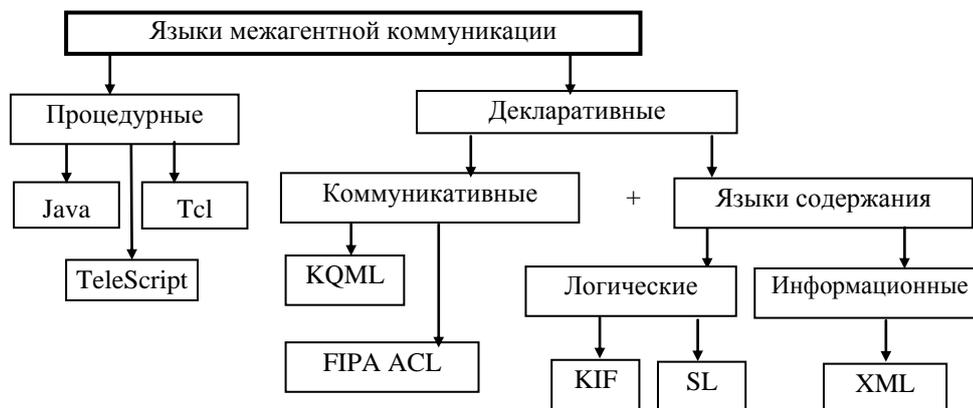


РИС. 3. Классификация языков межагентных коммуникаций

Известны два подхода к разработке таких языков – процедурный и декларативный. При первом подходе коммуникации происходят при выполнении инструкций. Язык проектируется с помощью Java или TCL (Tool Command Language).

При декларативном подходе коммуникации реализуются на основе описаний. Декларативный подход получил большое распространение для создания языков общения агентов, наиболее популярным из которых является KQML-структурированный язык взаимодействия агентов. Этот язык использу-

ется в качестве языка взаимодействия в различных многоагентных системах и средах для их программирования, таких как Agent-K, LALO, Java(tm) Agent Template (JATLite). Цели, аналогичные KQML, имеет и KAOs (Knowledgeable Agent-oriented System).

При создании межагентного сообщения необходим язык для представления самого содержания. Обычно для этого используют “логические языки”, представляющие знания как логические выражения (используя синтаксис, подобный LISP), и “информационных языков”, устанавливающих правила для описания типов информационных элементов.

Типичный логический язык содержания – это язык KIF (Knowledge Interchange Format), облегчающий обмен знаниями между системами искусственного интеллекта. Он использовался вместе с KQML в американских научно-исследовательских проектах по представлению знания в мультиагентных системах. Его синтаксис основан на Common LISP. Формат KIF является декларативным языком, который позволяет различным системам обмениваться онтологиями (соглашением между различными системами о структурах представления знаний различных предметных областей), используя тем самым в работе вычислительные преимущества этих систем.

Второй логический язык содержания – SL (Semantics Language), предложенный FIPA. SL-предложения выражаются логикой ментальных отношений и действий. Ментальная модель агента основана на представлении трех примитивов: убеждение, неопределенность, выбор. Основное свойство SL-логики позволяют смоделированным агентам находиться в соответствии с их ментальными отношениями.

В отличие от языков KIF и SL, язык XML не представляет информацию в виде логических выражений, а использует другие типы структур. XML-агенты обладают способностью по запросу предоставлять информацию из произвольных источников данных.

Стратегия выбора средств разработки интерфейсных агентов.

Анализ многочисленных публикаций и информации, представленной в электронном виде, показал, что она может быть структурирована, например, подобно рис. 1 – 3, а также табл. 1 – 4. Это, с одной стороны, позволяет создать информационный базис для построения онтологии предметной области – “интеллектуальные интерфейсы” как информационной основы для создания базы знаний, а с другой – предложить подход к выбору средств разработки интерфейсных агентов. При этом критерии для выбора должны определяться исходя из предметной области и назначения ИС. В качестве отправной позиции для выбора критериев можно использовать, например, следующий набор:

- возможность создания интерфейсных автономных агентов, способных интегрировать в Windows-приложения (критерий а);
- наличие операторов для временных выражений, поскольку агенты должны своевременно реагировать на действия (критерий б);

– поддержка архитектуры стиля BDI, так как предполагается наличие у интерфейсного агента знаний о желаниях, убеждениях, намерениях пользователя (критерий с);

– наличие операторов для реализации коммуникаций, поскольку интерфейсный агент должен быть способен к общению с пользователем (критерий d);

– пригодность языка для определения и программирования групп, поскольку предполагается работа в группе (критерий e);

– наличие конструкций для реализации модульностей из-за сложности программ интерфейсных агентов (критерий f);

– обеспечение четкой семантики (критерий g).

Однако, если для выбора языка программирования использовать только эти критерии (напрямую) для сравнения существующих языков, то выбрать среди них требуемый практически невозможно, поскольку каждый из них обладает почти одинаковыми свойствами (табл. 5).

ТАБЛИЦА 5. Сравнение языков программирования интерфейсных агентов по обозначенным выше критериям

Средство разработки агента / язык реализации	Критерии						
	a	b	c	d	e	f	g
Microsoft Agent / Visual Basic, Visual C++, Java, Delphi, Visual FoxPro (7.0)	+	+	-	-	+	+	+
Lotus Notes / LotusScript	-	+	-	+	+	+	+
NetStepper Pro 1.0. / Java	+	+	-	+	+	+	+
Agent Bulder. / Java	+	+	-	+	+	+	+
A. L. I. C. E.- /Java, AIML	+	+	-	+	+	+	+
AgentTool /Java	+	+	-	+	+	+	+
Grasshopper/ Java	+	+	-	+	+	+	+
JAFMAS/ Java	+	+	-	+	+	+	+
lyntin / Python	-	+	-	+	+	+	+
Narval /XML, Python	-	+	-	+	+	+	+

Поэтому авторы предложили подход к выбору средств разработки агентов, основанный на комплексной оценке параметров существующего множества средств и последующем отсеивании неприемлемых, исходя из требований к параметрам ИИС в целом, установления соответствия этим требованиям значений параметров проектируемого интерфейса системы, определения набора агентов, удовлетворяющих требованиям интерфейса, определения необходимого набора процедур и функций, которые должны реализовывать интерфейсные агенты, а также исходя из оценки параметров инструментальных средств и языка программирования по пользовательским критериям, таким как доступность, простота использования, наличие полного комплекта документации, стоимость и др.

Стратегия предлагаемого подхода к выбору средств для разработки агентов интеллектуального интерфейса ИИС заключается в выполнении такой последовательности этапов.

1. Определение назначения ИИС и требований к интеллектуальному интерфейсу, в первую очередь, к набору функций, которые должен реализовать интеллектуальный интерфейс.

2. Составление таблицы, аналогичной табл. 1, и определение ИИС-аналога с функциями интерфейса, близкими к функциям интерфейса рассматриваемой системы. Составление набора функций интеллектуального интерфейса для рассматриваемой ИИС.

3. Определение соответствия выделенных функций интерфейса ИИС, функциям, которые должны реализовать агенты, и способам реализации этих функций агентами (составление таблицы, аналогичной табл. 2).

4. Определение на основе данных табл. 1 и 2 требований к средствам разработки выделенного набора интерфейсных агентов, учитывая вышеприведенные критерии выбора, а также пользовательские характеристики (например, доступность для приобретения, простота эксплуатации и др.).

5. Выбор языка программирования агентов и определение набора инструментальных средств разработки агентов (по таблице, аналогичной табл. 3), учитывая определенные по п. 4 требования, а также наиболее значимые для области применения ИИС пользовательские характеристики. Составление таблицы параметров (аналогичной табл. 4) для выбранного набора средств.

6. Анализ параметров выбранных инструментальных средств разработки на соответствие вышеприведенным критериям выбора и требованиям, предъявляемым к интеллектуальному интерфейсу согласно п. 1, 4.

7. Уточнение параметров выбранного языка программирования агентов и инструментальных средств для их проектирования по данным аналога и дополнительной информации от фирмы поставщика или разработчика средств. Принятие решения по приобретению и использованию выбранного средства.

Следует подчеркнуть, что данный процесс выбора является итерационным, при этом скорость его сходимости существенно зависит от технических требований к ИИС в целом, от приложений, на которые ориентирована система, а также от квалификации разработчика интерфейса.

Таим образом, одним из перспективных направлений в развитии средств взаимодействия пользователя с ИИС является создание интеллектуального интерфейса с использованием агентных технологий. Существующее в настоящее время большое количество средств разработки программных агентов делает их выбор для конкретного использования весьма затруднительным. Предложенный подход для выбора средств создания интерфейсных агентов в определенной степени снижает трудоемкость решения проблемы выбора и создает исходный информационный базис для построения онтологии предметной области – “интеллектуальные интерфейсы ИИС” как основы создания в дальнейшем соответствующей базы знаний.

1. *Тельнов Ю.Ф.* Интеллектуальные информационные системы: Учеб. пособие. – М.: МЭСИ, 2001. – 81 с.
2. *Stuart E. Middleton.* Interface agents: A review of the field. Technical Report Number: ECSTR-IAM01-001 (http://www.ecs.soton.ac.uk/~sem99r/agent_survey.html).
3. *Steven Versteeg, Leon Sterling.* Languages for Mobile Agents. 433-463 Thesis. (<http://www.cs.mu.oz.au/~scv/thesis.html>).
4. *Term Paper Assignments.* Agent Construction Tools (<http://www.cs.odu.edu/~shen/intIIDB/Assignments.html>).
5. *General Software and Toolkits.* Agent-Based Computational Economics (ACE) and Complex Adaptive Systems (CAS) (<http://www.econ.iastate.edu/tesfatsi/acencode.htm>).
6. *Multiagent Systems.* Software: Tools for building MASs (http://www.multiagent.com/Software/Tools_for_building_MASs/index.html).
7. *Agent Programming Languages ~ An Overview* (http://students.csci.unt.edu/~croberts/5250/project/agent_programming.html).
8. *Autonomous Agents* (<http://rego96.ocs.lsu.edu/6.2/doc/HOWTOS/other-formats/html/AI-Alife-HOWTO-html/AI-Alife-HOWTO-6.html>).
9. *Замечательная технология MS AGENT и применение ее для программирования в среде VFP* (<http://www.stons.chat.ru/msagent.htm>).
10. *Горякин Р.* Программирование агентов (<http://kepten.fatal.ru/doci/delphi/rech/article.htm>).
11. *AgentBuilder Lite* An Integrated Toolkit for Constructing Intelligent Software Agents (<http://www.agentbuilder.com/Documentation/appnote/buyerSeller.html>).
12. *Dr. Richard S. Wallace.* Don't Read Me A. L. I. C. E. and AIML Documentation (<http://alicebot.org/articles/wallace/dont.html>).
13. *IBM Aglets.* Comparison of jafmas to other agent building tools (<http://www.ececs.uc.edu/~abaker/JAFMAS/tab3.html>).
14. *Christophe Vermeulen, Bart Bauwens.* Software Agents using XML for Telecom Service Modelling: a Practical Experience (<http://www.infoloom.com/gcaconfs/WEB/paris98/bauwens.htm>).

Получено 14.03.2005