

ТЕКУЩЕЕ СОСТОЯНИЕ И НАПРАВЛЕНИЯ РАЗВИТИЯ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Голенков В. В., Гулякина Н. А., Шункевич Д. В.

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: golen@bsuir.by, guliakina@bsuir.by, shunkevich@bsuir.by

В работе рассмотрены основные актуальные проблемы в области разработки современных интеллектуальных систем. Предложен подход к их решению, основанный на использовании открытой семантической технологии проектирования интеллектуальных систем

ВВЕДЕНИЕ

До настоящего времени традиционные информационные технологии и технологии искусственного интеллекта развивались независимо друг от друга.

Современное состояние информационных технологий в целом можно охарактеризовать как

- иллюзию благополучия;
- иллюзию всеислия финансовых ресурсов в решении различных технических задач;
- «вавилонское столпотворение» различных технических решений, о совместимости которых никто серьезно не задумывается;
- отсутствие комплексного системного подхода к автоматизации сложных видов деятельности;
- отсутствие осознания того, что недостатки современных информационных технологий имеют фундаментальный, системный характер.

Сейчас настало время фундаментального переосмысления опыта эволюции традиционных информационных технологий и их глубокой интеграции с технологиями искусственного интеллекта. Это необходимо для устранения целого ряда недостатков традиционных информационных технологий.

I. НЕДОСТАТКИ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

К недостаткам современных информационных технологий можно отнести:

1. Многообразие семантически эквивалентных форм (языков) представления (кодирования) обрабатываемой информации (знаний) в памяти компьютерных систем. Следствием указанного многообразия является многообразие семантически эквивалентных форм представления в памяти компьютерных систем способов решения задач (как процедурных, так и непроцедурных – функциональных, логических и т.д.). Отсутствие унификации представления различного вида знаний в памяти современных компьютерных систем приводит:

- к дублированию семантически эквивалентных информационных компонентов компьютерных систем;
 - к семантической несовместимости компьютерных систем и, следовательно, к высокой трудоемкости их интеграции в системы более высокого уровня иерархии;
 - к существенному снижению эффективности применения методики компонентного проектирования компьютерных систем на основе библиотеки многократно используемых компонентов (особенно, если речь идет о «крупных» компонентах, в частности, о типовых подсистемах).
2. Недостаточно высокую степень обучаемости современных компьютерных систем в ходе их эксплуатации, следствием чего является высокая трудоемкость их сопровождения и совершенствования, а также недостаточно длительный их жизненный цикл. Обучаемость компьютерной системы обеспечивается
 - гибкостью – многообразием и трудоемкостью возможных изменений, вносимых в систему в процессе пополнения системы новыми знаниями и навыками и совершенствования уже приобретенных знаний и навыков;
 - стратифицированностью системы – четким разделением системы на достаточно независимые друг от друга уровни иерархии;
 - рефлексивностью – способностью анализировать собственное состояние и свою деятельность;
 - гибридность – способностью приобретать и использовать широкое (а в идеале – неограниченное) многообразие знаний и навыков;
 - уровнем самообучаемости – уровнем автоматизации приобретения новых знаний и навыков, а также совершенствования уже приобретенных знаний и навыков

3. Отсутствие возможности у экспертов реально влиять на качество разрабатываемых компьютерных систем. Опыт разработки сложных компьютерных систем показывает, что посредничество программистов между экспертами и проектируемыми компьютерными системами существенно искажает вклад экспертов. При разработке компьютерных систем следующего поколения доминировать должны не программисты, а эксперты, способные точно излагать свои знания.
4. Отсутствие семантической (смысловой) унификации интерфейсной деятельности пользователей компьютерных систем, что вместе с многообразием форм реализации пользовательских интерфейсов приводит:
 - к серьезным накладным расходам на усвоение пользовательских интерфейсов новых компьютерных систем;
 - к неполному и неэффективному использованию возможностей эксплуатируемых компьютерных систем.

Преодолеть указанные недостатки можно только путем фундаментального переосмысления архитектуры и принципов организации сложных компьютерных систем. Основой такого переосмысления является устранение многообразия форм представления (кодирования) информации в памяти компьютерных систем.

Преодоление недостатков современных компьютерных систем предполагает:

- унификацию представления обрабатываемой информации
- функциональную унификацию (унификацию принципов обработки информации)

Важно отметить, что унификация (стандартизация) – это не ограничение творческой свободы инженера, а способ обеспечения совместимости результатов инженерной деятельности.

II. ПРИНЦИПЫ УНИФИКАЦИИ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ

Объективным ориентиром унификации представления информации в памяти компьютерных систем является формализация смысла представляемой информации.

Уточнение смыслового представления информации основано на максимально возможном упрощении синтаксиса при обеспечении универсальности.

Унификация информации, используемой в компьютерных системах, предполагает:

- синтаксическую унификацию используемой информации – унификацию формы представления (кодирования) этой информации. При этом следует отличать
 - кодирование информации в памяти компьютерной системы;

- представление информации для людей, обеспечивающее однозначность интерпретации (понимания, трактовки) этой информации разными людьми;
- семантическую унификацию используемой информации в основе которой лежит согласование и точная спецификация всех (!) используемых понятий (концептов) с помощью иерархической системы формальных онтологий.

Критерием эффективности синтаксической унификации представления информации любого вида в памяти компьютерных систем естественно считать максимально возможное упрощение синтаксиса путем исключения из такого внутреннего универсального языка всех средств, обеспечивающих коммуникационную функцию языка (т. е. обмен сообщениями). Для внутреннего языка компьютерной системы являются излишними коммуникационные возможности, такие как имена, союзы, предлоги, разделители, ограничители, склонения, спряжения и т. д.

Указанный внутренний универсальный язык компьютерных систем фактически есть не что иное, как язык смыслового представления знаний. Очевидно, что унификация формы представления информации в памяти компьютерных систем на основе формального представления ее смысла выглядит вполне логично, т. к. только смысл информации является единственной объективной основой для унификации ее представления (кодирования). Таким образом, формализация смысла – ключ к решению многих проблем разработки современных компьютерных систем.

Согласно В. В. Мартынову, «фактически всякая мыслительная деятельность человека (не только научная), как полагают многие ученые, использует внутренний семантический код, на который переводят с естественного языка и с которого переводят на естественный язык. Поразительная способность человека к идентификации огромного множества структурно различных фраз с одинаковым смыслом и способность запомнить смысл вне этих фраз убеждает нас в этом» [1].

Смысл – это абстрактная знаковая конструкция, являющаяся инвариантом максимального класса семантически эквивалентных знаковых конструкций (текстов), принадлежащих разным языкам и удовлетворяющая следующим требованиям:

- отсутствие синонимии знаков (многократного вхождения знаков с одинаковыми денотатами);
- отсутствие дублирования информации в виде семантически эквивалентных текстов (не путать с логической эквивалентностью);

- отсутствие омонимичных знаков (в том числе местоимений);
- отсутствие у знаков внутренней структуры (атомарный характер знаков);
- отсутствие склонений, спряжений (как следствие отсутствия у знаков внутренней структуры);
- отсутствие фрагментов знаковой конструкции, не являющихся знаками (разделителей, ограничителей, и т.д.);
- выделение знаков связей, компонентами которых могут быть любые знаки, с которыми знаки связей связываются синтаксически задаваемыми отношениями инцидентности.

Существенно подчеркнуть, что рассмотренные принципы смыслового представления информации приводят к нелинейным (графовым) знаковым конструкциям, что в некотором смысле усложняет организацию памяти компьютерных систем, но существенно упрощает их логическую архитектуру.

III. УНИФИКАЦИЯ ПРИНЦИПОВ ОБРАБОТКИ ИНФОРМАЦИИ

В качестве основы для унификации принципов обработки информации предлагается использовать многоагентный подход, который в настоящее время широко используется в различных областях. Ориентация на многоагентный подход как основу для организации обработки информации обусловлена следующими основными преимуществами такого подхода [2, 3]:

- автономность (независимость) агентов в рамках такой системы, что позволяет локализовать изменения, вносимые в систему при ее эволюции, и снизить соответствующие трудозатраты;
- децентрализация обработки, т.е. отсутствие единого контролирующего центра, что также позволяет локализовать вносимые в многоагентную систему изменения.

Однако, к основным недостаткам большинства популярных современных средств построения многоагентных систем можно отнести следующие:

- большинство современных средств построения многоагентных систем ориентированы на представление знаний агента при помощи узкоспециализированных языков, зачастую не предназначенных для представления знаний в широком смысле. Речь при этом идет как о знаниях агента о себе самом, так и о знаниях о внешней среде. В некоторых подходах вначале строится онтология, для создания которой, однако, часто используются средства с низкой выразительной способностью, не предназначенные для построения онтологий. В конечном итоге такой подход приводит к сильной

ограниченности возможностей разработанных многоагентных систем и их несовместимости;

- абсолютное большинство современных средств предполагает, что взаимодействие агентов осуществляется путем обмена сообщениями непосредственно от агента к агенту. Такой подход обладает существенным недостатком, связанным с тем, что в этом случае каждый агент системы должен иметь достаточно полную информацию о других агентах в системе, что приводит к дополнительным затратам ресурсов, кроме того добавление или удаление одного или нескольких агентов приводит к необходимости оповещения об этом других агентов;
- многие средства построения многоагентных систем построены таким образом, что логический уровень взаимодействия агентов жестко привязан к физическому уровню реализации многоагентной системы. Например, при передаче сообщений от агента к агенту разработчику многоагентной системы необходимо помимо семантически значимой информации указывать IP-адрес компьютера, на котором расположен агент-получатель, кодировку, с помощью которой закодирован текст сообщения и другую техническую информацию, обусловленную исключительно особенностями текущей реализации средств;
- в большинстве подходов среда, с которой взаимодействуют агенты, уточняется отдельно разработчиком для каждой многоагентной системы, что с одной стороны, расширяет возможности применения соответствующих средств, но с другой стороны приводит к существенным накладным расходам и несовместимости таких многоагентных систем. Кроме того, в ряде случаев разработчик также обязан учитывать особенности технической реализации средств разработки в плане их стыковки с предполагаемой средой, в роли которой может выступать, например, локальная или глобальная сеть.

В рамках данной работы перечисленные недостатки предполагается устранять за счет использования следующих принципов:

- коммуникацию агентов предлагается осуществлять по принципу «доски объявлений», однако, в отличие от классического подхода, в роли сообщений выступают спецификации в общей памяти компьютерной системы выполняемых агентами действий (процессов), направленных на решение каких-либо задач, а в роли среды коммуникации выступает сама эта память. Такой подход позволяет:

- исключить необходимость разработки специализированного языка для обмена сообщениями;
- обеспечить «обезличенность» общения, т.е. каждый из агентов в общем случае не знает, какие еще агенты есть в системе, кем сформулирован и кому адресован тот или иной запрос. Таким образом, добавление или удаление агентов в систему не приводит к изменениям в других агентах, что обеспечивает модифицируемость всей системы;
- агенты, в том числе, конечный пользователь, получают возможность формулировать задачи в *декларативном ключе*, т.е. не указывать для каждой задачи способ ее решения. Таким образом, агенту заранее не нужно знать, каким образом система решит ту или иную задачу, достаточно лишь специфицировать конечный результат;
- в роли внешней среды для агентов выступает та же общая память, в которой формулируются задачи и посредством которой осуществляется взаимодействие агентов. Такой подход обеспечивает унификацию среды для различных систем агентов, что, в свою очередь, обеспечивает их совместимость.
- спецификация каждого агента описывается средствами языка представления знаний в той же памяти, в которой осуществляется решение задач, что позволяет:
 - минимизировать число специализированных средств, необходимых для спецификации агентов, как языковых, так и инструментальных;
 - с одной стороны - минимизировать необходимую в общем случае спецификацию агента, которая включает условие его инициализации и программу, описывающую алгоритм работы агента, с другой стороны - обеспечить возможность неограниченного расширения такой спецификации для каждого конкретного случая;
- синхронизацию деятельности агентов предполагается осуществлять на уровне выполняемых ими процессов, направленных на решение тех или иных задач. Таким образом, каждый агент трактуется как некий абстрактный процессор, способный решать задачи определенного класса.
- каждый информационный процесс в любой момент времени имеет ассоциативный доступ к необходимым фрагментам базы знаний, хранящейся в общей памяти, за исключением фрагментов, заблокированных другими процессами в соответствии с исполь-

зуемым механизмом синхронизации выполнения параллельных процессов. Таким образом, с одной стороны, исключается необходимость хранения каждым агентом информации о внешней среде, с другой стороны, каждый агент, как и в классических многоагентных системах, обладает только частью всей информации, необходимой для решения задачи.

Важно отметить, что в общем случае невозможно априори предсказать, какие именно знания, модели и способы решения задач понадобятся системе для решения конкретной задачи. В связи с этим необходимо обеспечить, с одной стороны, возможность доступа ко всем необходимым фрагментам базы знаний (в пределе - ко всей базе знаний), с другой стороны - иметь возможность локализовать область поиска пути решения задачи, например, рамками одной предметной области.

IV. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТРЕБОВАНИЯ К ИХ РАЗРАБОТЧИКАМ

Таким образом, устранение проблем современных информационных технологий фактически преобразует современные компьютерные системы в интеллектуальные системы. Следовательно, интеллектуальные системы – это не альтернативная ветвь развития компьютерных систем, а естественный этап их эволюции, направленной на повышение уровня гибкости, стратифицированности, рефлексивности, гибридности, и обучаемости компьютерных систем.

С учетом сказанного, можно определить *интеллектуальную систему* как компьютерную систему, основанную на знаниях, обладающую высоким уровнем обучаемости, т.е. способностью быстро приобретать новые и совершенствовать уже приобретенные знания и навыки и при этом не иметь никаких ограничений на вид приобретаемых и совершенствуемых ею знаний и навыков, а также на способ их совместного использования.

Важно подчеркнуть, что переход от проектирования традиционных компьютерных систем к проектированию интеллектуальных систем требует от разработчиков существенного повышения:

- уровня математической культуры при построении формальной модели среды, в которой функционирует интеллектуальная система, формальных моделей решаемых ею задач и формальных моделей различных используемых ею способов решения задач;
- уровня системной культуры, позволяющей адекватно оценивать качество разрабатываемых систем с точки зрения общей теории систем и, в частности, оценивать об-

щий уровень автоматизации, реализуемый с помощью этих систем;

- уровня технологической культуры, обеспечивающей совместимость разрабатываемых систем, а также постоянное расширение библиотеки многократно используемых компонентов создаваемых систем.

V. РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННЫХ ПРИНЦИПОВ УНИФИКАЦИИ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ

Рассмотренные принципы унификации представления и обработки информации реализуются в виде открытой семантической технологии компонентного проектирования гибких совместимых гибридных компьютерных систем нового поколения, названной нами Технологией OSTIS (Open Semantic Technology for Intelligent Systems) [4, 5].

Перечислим основные принципы предлагаемой семантической технологии:

- ориентация на смысловое однозначное представление знаний в виде семантических сетей, имеющих базовую теоретико-множественную интерпретацию, что обеспечивает решение проблемы многообразия форм представления одного и того же смысла, и проблемы неоднозначности семантической интерпретации информационных конструкций;
- использование ассоциативной графодинамической памяти;
- применение агентно-ориентированной модели обработки знаний;
- реализация предлагаемой технологии в виде интеллектуальной Метасистемы IMS.ostis [6], которая построена по этой же технологии и осуществляет поддержку проектирования компьютерных систем, разрабатываемых по указанной технологии;
- обеспечение в проектируемых системах высокого уровня гибкости, стратифицированности, рефлексивности, гибридности и, как следствие, обучаемости.

В основе предлагаемой семантической технологии лежит базовый унифицированный язык смыслового кодирования информации в памяти компьютерных систем, удовлетворяющий перечисленным ранее требованиям и названный **SC-кодом** (Semantic Computer Code). Тексты SC-кода (sc-тексты) представляют собой семантические сети с базовой теоретико-множественной интерпретацией. Элементы таких сетей называются *sc-элементами* (sc-узлами, sc-дугами, sc-ребрами). Теоретико-множественная интерпретация предполагает, что каждый sc-элемент может быть либо знаком множества sc-элементов, либо знаком терминальной сущности (сущности, не являющейся множеством). В свою очередь,

sc-дуги и sc-ребра трактуются как знаки двухмощных множеств (ориентированных или неориентированных соответственно), элементами которых являются sc-элементы, инцидентные соответствующей sc-дуге или sc-ребру. При этом выделяется класс *sc-дуг основного вида*, каждая из которых трактуется как принадлежность элемента множеству. Таким образом, SC-код как язык семантических сетей объединяет два базовых аспекта представления информации – синтаксический (выделение синтаксических типов sc-элементов – sc-дуг, sc-узлов и т.д.), и семантический (выделение базовой типологии сущностей, обозначаемых sc-элементами, – терминальных сущностей, множеств, отношений и т.д.).

Системы, построенные на основе Технологии OSTIS, называются ostis-системами. Каждая ostis-система в общем случае имеет архитектуру, представленную на рисунке 1.

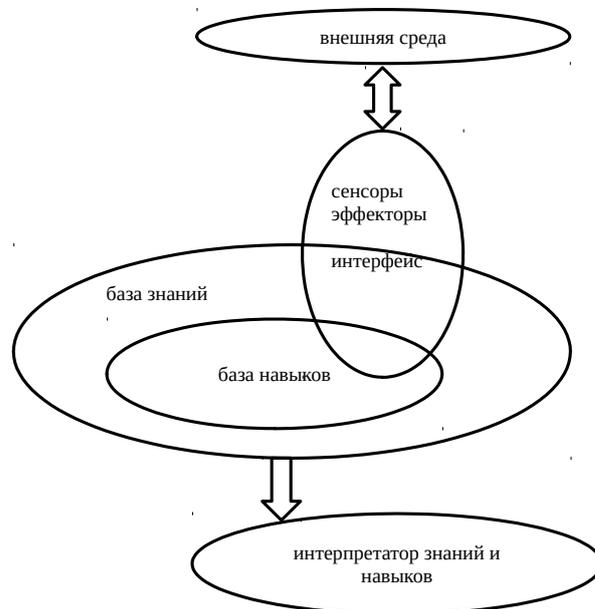


Рис. 1 – Архитектура ostis-системы

Под **базой знаний** ostis-системы понимается систематизированная совокупность всех знаний и навыков, хранимых в памяти такой системы. Как видно из рисунка, частью базы знаний является **база навыков**, то есть систематизированный комплекс всех известных интеллектуальной системе способов решения различных задач и классов задач. Возможность описания в базе знаний способов решения различных задач достигается за счет:

- наличия в технологии формальных средств описания в базе знаний алгоритмов, программ любого уровня сложности (как императивных, так и декларативных);
- описания в базе знаний системы спецификаций всех агентов обработки знаний;
- описания в базе знаний системы программ агентов обработки знаний, что становится

возможным благодаря наличию в рамках технологии графового процедурного языка программирования SCP (Semantic Code Programming), который, с одной стороны, ориентирован на обработки sc-текстов, а с другой стороны, тексты программ которого записываются в SC-коде, т.е. являются sc-текстами.

Таким образом, каждая ostis-система состоит из полной модели этой системы, описанной средствами SC-кода и хранящейся в базе знаний (sc-модели ostis-системы) и интерпретатора знаний и навыков (платформы интерпретации sc-моделей ostis-систем), который в общем случае может быть реализован как программно, так и аппаратно. При этом если фундаментальной основой традиционных информационных технологий и соответствующих ей компьютерных архитектур является абстрактная машина фон Неймана, то основой ostis-систем как нового этапа эволюции компьютерных систем должны стать семантические параллельные компьютеры с нелинейной ассоциативной реконфигурируемой памятью. Это очевидное направление повышения производительности компьютерных систем нового поколения.

Интерпретатор знаний и навыков включает такие обязательные компоненты как хранилище sc-текстов (sc-хранилище), обеспечивающее хранение базы знаний и доступ к ней, а также интерпретатор программ языка SCP (scp-программ).

Приведенная архитектура позволяет обеспечить платформенную независимость ostis-систем, в том числе возможность легкого перевода разработанной системы с программной платформы на аппаратную.

Интерфейс ostis-системы в рамках предлагаемого подхода трактуется как подсистема, которая строится по тем же принципам, то есть имеет свою специализированную базу знаний и базу навыков.

Достоинства предлагаемой семантической технологии заключаются в том, что она:

- ориентирована на разработку компьютерных систем нового поколения (гибридных и семантических совместимых компьютерных систем с высокой степенью обучаемости);
- имеет открытый характер как для ее пользователей (разработчиков прикладных интеллектуальных систем), так и для тех, кто желает участвовать в ее совершенствовании;
- ориентирована на постоянное повышение темпов ее эволюции;
- является основой для решения проблем семантической совместимости самых раз-

личных научных и технических знаний, так как она ориентирована на формализацию междисциплинарных связей самого различного вида.

ЗАКЛЮЧЕНИЕ

В работе рассмотрены основные недостатки современных компьютерных систем, предложены подходы к их устранению, реализуемые в виде открытой семантической технологии проектирования интеллектуальных систем.

Следует подчеркнуть, что исследования в области искусственного интеллекта носят ярко выраженный междисциплинарный характер. О необходимости глубокой конвергенции различных научных направлений в области искусственного интеллекта в целях построения общей формальной теории интеллектуальных систем говорится в целом ряде работ, в частности, Янковской А.Е. [7] и Палагина А.В. [8]. Этому также посвящены конференции «Artificial General Intelligence», проводимые с 2008 года [9]. Основой для построения такой теории могут стать предлагаемые в работе принципы унификации представления и обработки информации и построенная на их основе технология проектирования интеллектуальных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Мартынов, В.В. Семиологические основы информатики / В.В. Мартынов - Мн.: Наука и техника, 1974. - 192 с.
2. Wooldridge, M. An introduction to multiagent systems / M. Wooldridge. - 2nd ed. - Chichester : J. Wiley, 2009. - 484 p.
3. Тарасов, В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика / В. Б. Тарасов. - М. : Эдиториал УРСС, 2002. - 348 с.
4. Голенков, В. В. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания / В. В. Голенков, Н. А. Гулякина // Онтология проектирования. - 2014. - № 1. - С. 42-64.
5. Голенков, В. В. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 2: Унифицированные модели проектирования / В. В. Голенков, Н. А. Гулякина // Онтология проектирования. - 2014. - № 4. - С. 34-53.
6. Метасистема IMS.ostis [Электронный ресурс]. - Режим доступа: <http://ims.ostis.net>. - Дата доступа: 10.09.2018.
7. Янковская, А.Е. Анализ данных и знаний на основе конвергенции нескольких наук и научных направлений / А.Е. Янковская // Международная конференция «Интеллектуализация обработки информации» (ИОИ-8), Кипр, г. Пафос, 17-24 октября 2010 г. - сс. 196-199.
8. Палагин, А.В. Проблемы трансдисциплинарности и роль информатики / А.В. Палагин // Кибернетика и системный анализ. - 2013. №5. - сс. 3-13.
9. Конференция Artificial General Intelligence (AGI) <http://agi-conference.org>