УДК 004.4'2+82+75+771

Дата подачи статьи: 10.10.17

DOI: 10.15827/0236-235X.123.527-536

2018. Т. 31. № 3. С. 527–536

Облачная платформа IASPaas для разработки оболочек интеллектуальных сервисов: состояние и перспективы развития

*В.В. Грибова*¹, д.т.н., старший научный сотрудник, зам. директора, gribova@iacp.dvo.ru

*А.С. Клещев*¹, д.ф.-м.н., профессор, главный научный сотрудник, kleschew@iacp.dvo.ru

*Ф.М. Москаленко*¹, к.т.н., старший научный сотрудник, philipmm@iacp.dvo.ru

*В.А. Тимченко*¹, к.т.н., старший научный сотрудник, vadim@iacp.dvo.ru

*Л.А. Федорищев*¹, к.т.н., научный сотрудник, leo@iacp.dvo.ru

*Е.А. Шалфеева*¹, к.т.н., доцент, старший научный сотрудник, shalf@iacp.dvo.ru

¹ Институт автоматизации и процессов управления ДВО РАН, г. Владивосток, 690041, Россия

В работе описаны основные характеристики и функциональные возможности платформы облачных вычислений IASPaas, которая поддерживает три модели предоставления облачных сервисов – PaaS, SaaS и DaaS. Платформа предназначена для разработки специализированных (то есть ориентированных на определенные предметные области и/или классы решаемых задач) оболочек интеллектуальных сервисов, а также прикладных интеллектуальных сервисов с использованием таких оболочек, которые при этом предоставляются и как облачные сервисы платформы. Создание (сопровождение) прикладных сервисов с помощью оболочки сводится к формированию (модификации) базы знаний с использованием предоставляемых этой оболочкой средств формирования баз знаний и связыванию ее с решателем задач. Решатель задач состоит из набора агентов – программных компонентов, взаимодействующих друг с другом посредством обмена сообщениями.

В специализированных оболочках при представлении знаний используется специфичное для предметной области концептуальное представление, определяемое онтологией той области, для которой создается оболочка. Такая проблемно-ориентированная модель (язык) представления знаний используется средством формирования баз знаний для генерации ориентированного на экспертов предметной области пользовательского интерфейса. В результате эксперты могут формировать и сопровождать базы знаний и данных в привычной для себя системе понятий (без посредников в лице инженеров-когнитологов или дополнительного обучения), не совершая ошибок в использовании языка представления знаний.

Платформа IASPaas предоставляет базовую (универсальную) и множество специализированных технологий разработки оболочек прикладных интеллектуальных сервисов с использованием инструментальных средств поддержки этих технологий. Также предоставляется технология взаимодействия агентов решателей задач с внешним (не входящим в состав платформы IASPaas) программным обеспечением. Она основана на стандартном механизме обработки http-запросов и возможности запуска исполняемых файлов внешнего программного обеспечения из программ (скриптов), расположенных на web-сервере.

Ключевые слова: оболочки экспертных систем, интеллектуальные системы, технология разработки интеллектуальных систем, облачная платформа, облачные вычисления, интеллектуальные сервисы, мультиагентные системы.

Технологии облачных вычислений имеют сегодня широкое распространение. Рынок предоставляемых облачных решений и услуг достаточно разнообразен и предлагает как отдельные компоненты и услуги, так и широкий спектр комплексных решений, реализующих несколько моделей, – IaaS, PaaS, SaaS, DaaS и др. Анализируя имеющиеся на рынке облачные инфраструктуры, предлагающие средства разработки различных приложений – модель PaaS [1–3], можно констатировать, что все они предназначены для решения широкого круга задач и содержат системы управления БД (MySQL, Redis, MongoDB, PostgreSQL, Memcached, Oracle и др.), а также средства разработки приложений на объектно-ориентированных и скриптовых языках программирования (Java, PHP, Python, Ruby, JavaScript, Node.js, Go, .NET и др.).

На сегодняшний день актуальной является задача поиска методов развития уровня PaaS, позволяющих упростить создание сложных прикладных

сервисов разных типов на основе масштабируемых облачных сред. Решению данной задачи посвящено достаточно большое количество проектов, и многие облачные платформы имеют специализированный инструментарий для поддержки разработки приложений разных типов. Типичным примером являются универсальные и специализированные инструменты разработки web-сайтов. В то же время большой класс задач, решение которых возможно лишь с помощью *интеллектуальных систем* (ИС), в архитектуре которых присутствует специализированный компонент – база знаний, не покрывается этими проектами. Общеизвестно, что универсальные технологии либо полностью не применимы для создания ИС, либо значительно усложняют процесс их разработки и, прежде всего, сопровождения; вследствие этого жизнеспособность таких систем оказывается низкой [4].

Для снижения трудоемкости разработки и сопровождения ИС традиционно используются обо-

лочки, каждая из которых предлагает разработчику систему управления базами знаний, основанную на некоторой модели представления знаний, и решатель задач, реализующий алгоритм обработки знаний и данных [4–7]. Создание (сопровождение) ИС с помощью оболочки сводится к формированию (модификации) базы знаний через систему управления базами знаний и связыванию ее с решателем задач. Разработка оболочек является отдельной и сложной проблемой; в настоящее время неизвестны облачные платформы, которые поддерживают их разработку.

Целью данной работы является описание облачной платформы IACPaaS, предназначенной для разработки оболочек интеллектуальных сервисов, ее основных принципов, архитектуры и функциональных возможностей.

Основные принципы, положенные в основу платформы IACPaaS

Платформа IACPaaS (далее – платформа) предназначена для разработки оболочек интеллектуальных сервисов по нескольким технологиям, состав которых является расширяемым, создания с помощью этих оболочек прикладных интеллектуальных сервисов, а также использования этих сервисов для решения задач.

Опишем общие принципы, положенные в основу платформы IACPaaS.

Поддерживаемые типы оболочек. Большинство оболочек, реализованных к настоящему времени, являются проблемно-независимыми. Их достоинство в универсальности – потенциальной возможности использования для создания ИС, предназначенных для решения широкого круга задач в различных предметных областях. Стремление разрабатывать именно проблемно-независимые оболочки с возможно более широкой областью применения связано и с высокой трудоемкостью их реализации. В то же время разработка и сопровождение ИС с помощью проблемно-независимой оболочки, не учитывающей специфику конкретной предметной области и задачи, часто оказываются трудоемкими и интеллектуально сложными.

Проблемно-ориентированные (или специализированные) оболочки, как правило, имеют более узкую область применения, чем проблемно-независимые. Однако создание и сопровождение ИС для задач и предметных областей, на которые ориентирована такая оболочка, имеют меньшую трудоемкость и интеллектуальную сложность, что делает такие ИС более жизнеспособными [8–10].

Платформа IACPaaS ориентирована на снижение трудоемкости разработки оболочек, прежде всего проблемно-ориентированных, что, в конечном счете, имеет целью повышение жизнеспособности интеллектуальных сервисов, создаваемых с помощью таких оболочек.

Модель представления знаний и данных. Системы управления базами знаний проблемно-независимых оболочек основаны на проблемно-независимых моделях представления знаний и данных – системах продукций (правил), фреймах, объектно-ориентированных моделях [11, 12]. В системах продукций предметные и процедурные знания неразрывно связаны и имеют форму правил; в системах фреймов процедурные знания присоединяются к предметным, но имеют иное (процедурное) представление; в объектно-ориентированных моделях представляются только предметные знания и данные. Хотя практика показала, что системы управления базами знаний, основанные на проблемно-независимых моделях представления знаний и данных, позволяют формировать небольшие базы знаний в различных предметных областях, формирование и особенно сопровождение больших (компетентных) баз знаний – задача чрезмерной интеллектуальной сложности. Причина в том, что база знаний создается экспертами в соответствующей предметной области, а не специалистами в области инженерии знаний или программирования. Эксперту для формирования базы знаний и ее сопровождения необходимо, чтобы знания имели привычную для него форму и ему не приходилось бы преобразовывать знания в форму, навязываемую моделью представления знаний.

Система управления базами знаний проблемно-ориентированной оболочки основана на некоторой проблемно-ориентированной модели представления знаний и данных. В такой модели представляются только предметные знания и данные; она учитывает специфику организации знаний и данных в конкретной предметной области. Благодаря этим свойствам только системы управления базами знаний, основанные на проблемно-ориентированных моделях представления знаний, снижают интеллектуальную сложность формирования и сопровождения баз знаний до приемлемой. Вместе с тем класс таких моделей весьма широк; в качестве примеров можно указать модели представления медицинских знаний о диагностике и лечении [13], математических знаний и знаний о способах рассуждений при доказательстве теорем, знаний о компьютерных программах и их преобразованиях [14], знаний в области неорганической и органической химии [15], знаний о виртуальных мирах и сценах [16], знаний в области технической диагностики роботов [17].

Платформа IACPaaS предлагает разработчикам оболочек интеллектуальных сервисов универсальный проблемно-независимый метаязык для спецификации абстрактного синтаксиса моделей (языков) представления знаний [18]. Метаязык позволяет определить абстрактный синтаксис той модели (языка) представления знаний, которая учитывает организацию знаний и данных конкретной предметной области.

Система управления базами знаний. Система IACPaas основана на метаязыке для спецификации абстрактного синтаксиса моделей представления знаний. С помощью специального структурного редактора, входящего в состав IACPaas [19], инженер знаний может определить абстрактный синтаксис необходимой модели представления знаний в форме грамматики, порождающей размеченные графы [20, 21], – семантические представления баз знаний и данных. Сама грамматика также имеет форму графа с единственной выделенной начальной вершиной (аксиомой). По этой грамматике эксперт с помощью структурного редактора может формировать и модифицировать базы знаний и данных. Структурный редактор генерирует для него полимодальный интерфейс – база знаний и данных (и их разные фрагменты) может представляться в виде размеченного квазидерева, графа, текста, таблицы.

Решатели задач. Разрабатываемые средствами платформы IACPaas решатели задач имеют формальные параметры, каждый из которых состоит из имени формального параметра и грамматики, задающей абстрактный синтаксис модели представления знаний или данных, средствами которой представляются фактические параметры, соответствующие этому формальному [22]. Создание интеллектуального сервиса на основе оболочки состоит в формировании с помощью системы управления базами знаний необходимых баз знаний и данных – фактических параметров, соответствующих формальным параметрам решателя, и сборке из них и решателя интеллектуального сервиса. Решатели задач разрабатываются на основе агентного подхода, то есть решатель задач состоит из множества агентов, которые обмениваются сообщениями. Такой подход обеспечивает гибкость при создании решателей: решатель может «собираться» из агентов, разработанных ранее.

Использование внешних модулей для создания ИС. Тенденции развития программного обеспечения показывают, что во многих случаях (например, для реализации динамических экспертных систем) к ИС требуется подключение различных дополнительных модулей, реализованных на сторонних платформах. Платформа IACPaas обеспечивает двусторонний обмен данными между решателями задач платформы IACPaas и сторонними компонентами на других платформах.

Реализация совместной разработки ИС и их компонентов. Необходимость включения в процесс разработки оболочек и интеллектуальных сервисов специалистов разного профиля (инженеров знаний, экспертов предметной области, проектировщиков, программистов, дизайнеров интерфейса и др.) требует организации их совместной работы. Платформа обеспечивает возможность коллективной разработки оболочек и интеллектуальных сервисов через личные кабинеты разработчиков и механизмы управления процессом разработки.

Поддержка различных технологий создания оболочек. Инструментарий для разработки приложений на платформе IACPaas можно разделить на два основных типа – универсальный и ориентированный на проблему либо задачу. Универсальный инструментариий позволяет создавать решатели задач любого типа, но процесс создания является более трудоемким по сравнению с использованием специализированного инструментариия. Платформа поддерживает создание оболочек по нескольким технологиям: базовой, поддерживаемой универсальными средствами разработки, и специализированным, поддерживаемым специализированными средствами разработки. Набор технологий допускает расширение.

Модель представления знаний и данных. Система управления базами знаний

Обязательным компонентом любой оболочки ИС является система управления базами знаний. Если в универсальных оболочках базы знаний описываются на специализированных универсальных языках, поддерживающих известные модели представления знаний, то в проблемно-ориентированных оболочках при представлении знаний используется специфичное для предметной области концептуальное представление, определяемое онтологией той области, для которой создается оболочка.

Для спецификации абстрактного синтаксиса необходимой модели представления знаний используется универсальный метаязык, входящий в состав платформы. Сформированный с его помощью абстрактный синтаксис модели представления знаний – это пара $\langle GM, \sigma M \rangle$, где GM – граф понятий, в терминах которого формируются базы знаний и данных, возможно, содержащий циклы и петли, а σM – разметка этого графа. Граф GM – это тройка $GM = \langle Vertices, Arcs, RootVertex \rangle$, где $Vertices$ – множество вершин графа, $Arcs$ – множество дуг графа, $RootVertex$ – корневая вершина графа ($RootVertex \in Vertices$). Средства разметки σM включают в себя разметку вершин σV и разметку дуг σA . Эта разметка позволяет при описании абстрактного синтаксиса необходимой модели представления знаний задать ограничения на структуру и содержание знаний или данных конкретной предметной области. Подробно метаязык описания моделей представления знаний рассмотрен в [18].

Базы знаний и данных, сформированные по описанной модели представления знаний, также являются парой $KB = \langle GI, \sigma I \rangle$, где GI – граф (сеть) понятий, σI – разметка этого графа. Граф GI – это тройка $GI = \langle IVertices, IArcs, IRootVertex \rangle$, компоненты которой представляют собой множество вершин ($IVertices$), дуг ($IArcs$) и корневую вершину ($IRootVertex \in IVertices$). Средства разметки σI , в

отличие от σM , включают в себя только разметку вершин, то есть $\sigma I \equiv \sigma V$. Каждая вершина графа информации отображается в пару $\sigma V = \langle SM, RM \rangle$, где SM – служебная метка, RM – метка соответствия между моделью представления знаний (данных) и ее концептуальным представлением. Метка соответствия RM – это имя понятия предметной области, которому соответствует либо имя понятия предметной области, заданное в концептуальной модели, либо некоторое имя, входящее в класс предметных областей, также заданный в этой концептуальной модели.

Соответствие между базами знаний (данных) и их концептуальными представлениями задается следующим образом [23]. Каждой вершине в графе, представляющем базу знаний (данных), соответствует некоторая единственная вершина в ее концептуальной модели, называемая вершиной-прототипом (и каждой вершине в графе, представляющем концептуальную модель, соответствует некоторое множество вершин в базе знаний (данных)). Количество вершин в базе знаний (данных) определяется разметкой графа, представляющего его концептуальную модель. Проводя аналогию с программированием, можно заметить, что в предлагаемой модели, с одной стороны, вся информация представляется в едином унифицированном формате – в форме корневых иерархических орграфов, в чем можно усмотреть прямое сходство с функциональным программированием, в котором все данные представлены также унифицированно – списками, с другой стороны, в традиционном программировании общепринятой практикой является разделение информации (входной, промежуточной и выходной) на множество разрозненных информационных единиц, соответствующих типам данных (программа имеет множество переменных различных типов). В связи с этим программист должен знать все неявные связи между переменными и выражать их через алгоритмы обработки. Предлагаемая модель основана на обратном принципе: количество орграфов невелико, они имеют более сложную структуру, но эта структура явно определяет связи между всеми информационными единицами.

Внутреннее представление баз знаний, порожаемых на основе концептуальной модели представления знаний, также имеет вид корневых иерархических орграфов (с возможными петлями и циклами). Терминальные вершины этих орграфов представляют объекты разных сортов: строки, целочисленные и вещественные константы и др. Иерархичность орграфа означает, что его вершина может быть либо простой (не имеющей внутренней структуры), либо некоторым другим орграфом (или его подграфом).

Для управления (формирования, модификации, отладки) баз знаний и данных платформа содержит специализированный редактор [19], который автоматически генерирует пользовательский интер-

фейс по модели представления знаний. Для представления базы знаний (данных) в удобной для пользователей форме (модели представления) платформа имеет дополнительные средства генерации мультимодальных интерфейсов в форме графов, текстов и таблиц.

Решатели задач

Решатели задач разрабатываются на основе агентного подхода: решатель задач состоит из множества обменивающихся сообщениями агентов. Среди агентов решателя выделяются *корневой* (начинает работать первым), *интерфейсный контроллер* (обеспечивает сопряжение пользовательского интерфейса с решателем задач) и множество *обрабатывающих* агентов. Разработка кода агентов осуществляется на языке Java. Пользовательский web-интерфейс может быть создан несколькими способами: с использованием набора интерфейсных элементов, поддерживаемых специализированным интерфейсным агентом «Вид», а также средствами JavaScript и Flash. Разработка решателей задач подробно описана в [22].

Каждый решатель задач представляется корневым орграфом, который порожден по хранящейся в Фонде платформы метаинформации решателей задач (см. <http://www.swsys.ru/uploaded/image/2018-3/2018-3-dop/11.jpg>), интерпретируемой как абстрактный синтаксис языка спецификации решателей задач.

Агент декларативно представляется корневым орграфом, который порожден по хранящейся в Фонде платформы метаинформации агентов (см. <http://www.swsys.ru/uploaded/image/2018-3/2018-3-dop/12.jpg>), интерпретируемой как абстрактный синтаксис языка спецификации агентов. Некоторые терминальные вершины сети, описывающей агента, представлены объектами сорта бинарные данные, содержащими как исходный, так и исполняемый код агента [24].

Каждый шаблон сообщений является метаинформацией, интерпретируемой как абстрактный синтаксис языка сообщений агентов. Некоторые терминальные вершины орграфа, описывающего шаблон сообщений, также могут быть представлены объектами сорта бинарные данные, содержащими методы создания и обработки сообщений [24].

Представление всех данных платформы в форме корневых иерархических орграфов обеспечивает универсальность их обработки с помощью единого программного интерфейса (API) для доступа к единицам хранения всех типов, а также единого инструментального сервиса платформы – структурного редактора (для их формирования и сопровождения). Хранимые на платформе данные могут быть экспортированы во внешние форматы, количество которых расширяемо.

Использование внешних модулей для создания ИС

Платформа поддерживает создание оболочек и сервисов на гетерогенных вычислительных архитектурах: одна часть компонентов сервиса может быть создана непосредственно на платформе IASaaS (с использованием оболочек), другая часть – на сторонних платформах. При создании решателя задач оболочки также возможно его обращение к сторонним модулям на основе http-протокола. Для реализации механизма отправки http-запросов из программных агентов платформы на внешние адреса был разработан агент-посредник. Агенты платформы взаимодействуют не напрямую с внешними модулями, а с агентом-посредником (посредством стандартного обмена сообщениями).

Реализация совместной разработки ИС и их компонентов

Каждому зарегистрированному пользователю платформы предоставляется виртуальное рабочее место – личный кабинет. Личные кабинеты представляют собой набор инструментов управления личными данными и сервисами пользователя.

Пользователи платформы разделяются на следующие типы: гость, зарегистрированный пользователь и служебный пользователь. Гостю доступна ограниченная функциональность, он может просмотреть содержимое Фонда платформы; зарегистрировавшись на платформе, он становится зарегистрированным пользователем, ему предоставляются личный кабинет и набор функциональных возможностей: запуск сервисов, создание оболочек сервисов по предлагаемым технологиям, создание сервисов с использованием оболочек, создание новых работ и технологий.

Зарегистрированные пользователи могут иметь дополнительные служебные роли: управляющий единицей хранения, администратор раздела, руководитель групповой разработки компонентов оболочки либо сервиса на ее основе. Руководитель группы определяет и приглашает участников созданной группы, распределяет роли, назначает сроки исполнения, а также формирует (передает) необходимые для выполнения работы ресурсы. Помимо перечисленных служебных ролей зарегистрированных пользователей, у платформы есть администратор, который отвечает за набор разделов Фонда платформы в целом, назначает администраторов разделов, удаляет пользователей, если они нарушают политику платформы.

Поддержка различных технологий создания оболочек

Пользователи могут разрабатывать оболочки ИС (возможно создание ИС с нуля, без использова-

ния оболочек, что в общем случае значительно более трудоемко) в соответствии с *работами*, поддерживаемыми платформой, а также создавать новые работы. Под работой понимается набор инструментальных сервисов, с помощью которых можно создавать или модифицировать компоненты (определенного типа) разрабатываемой ИС или оболочки. Работа может быть простой или составной. Составная работа включает несколько этапов (шагов). Примером простой работы является создание базы знаний на основе ее концептуальной модели (пустой базы знаний). Пример составной работы – разработка агента, которая включает разработку шаблонов сообщений (этап пропускается, если все необходимые шаблоны сообщений содержатся в Фонде платформы) и декларативной части агента, генерацию заготовки кода агента на языке Java, его загрузку на платформу и интеграцию в декларативную спецификацию агента.

Компоненты ИС и оболочки создаются в соответствии с *технологиями*, доступными на платформе. Технология описывает работы по созданию оболочек (либо ИС) и частичный порядок выполнения этих работ. Технологии, поддерживаемые платформой, разделены на два класса: *базовая технология* и множество *специализированных технологий*.

Базовая технология позволяет создать ИС и оболочку для любого класса решаемых задач, при этом архитектура решателя допускает разные способы разбиения на агенты и их компоновки. Подробное описание технологии дано в [22, 24]. В настоящее время с использованием базовой технологии разработан ряд ИС и оболочек, а также инструментальных сервисов платформы, например, редактор графических сцен, интерпретатор виртуальных сред, оболочка для медицинской диагностики, сервис по диагностике острых конъюнктивитов и др.

Специализированные технологии ориентированы либо на конкретный класс задач, либо на определенную архитектуру решателя задач. В любом случае архитектура решателя задач такой оболочки или сервиса имеет заданный технологией состав компонентов. Количество специализированных технологий может расширяться пользователями платформы. Каждая технология поддерживается набором инструментальных сервисов и информационных ресурсов, необходимых для создания оболочки или сервиса. В настоящее время платформа поддерживает создание ИС и оболочек по двум специализированным технологиям: технологии разработки ИС и оболочек с интерактивными виртуальными средами и технологии разработки ИС и оболочек с агентами, присоединенными к метайнформации.

Специализированная технология разработки ИС и оболочек с интерактивными виртуальными средами предназначена для создания облачных

сервисов с 3D-сценами. В соответствии с данной технологией оболочка содержит онтологию виртуальной сцены, по которой описывается 3D-сцена, причем описание сцены разделено на два этапа – логическое и презентационное. Логическое описание формирует эксперт предметной области с помощью редактора информации, презентационное – дизайнер с помощью графического редактора, разработанного для поддержки данной технологии. Сформированное описание 3D-сцены подается на вход интерпретатору виртуального мира (решателю задач, разработанному для поддержки данной технологии), который затем автоматически генерирует 3D-сцену и обеспечивает управление ею в процессе работы пользователя с облачным сервисом. Более подробное описание технологии и реализованных сервисов дано в работе [16].

Специализированная технология разработки сервисов и оболочек с агентами, присоединенными к метainформации, удобна для использования в двух случаях: если результатом работы сервиса является некоторый выходной информационный ресурс, формирование которого по его концептуальной модели осуществляется методом «сверху-вниз»; если разработчик может декларативно (в виде информационного ресурса) описать решатель задач. В этом случае решатель задач – это специальный инструментальный сервис «расширяемый редактор информации», управляемый метainформацией либо выходного информационного ресурса, либо управляющей структуры, к некоторым вершинам которой присоединены агенты, порождающие фрагменты выходного информационного ресурса (результата работы сервиса) или вы-

полняющие другие требуемые для работы сервиса преобразования. С помощью данной технологии разработана оболочка для исследований в области верификации интуитивных математических доказательств.

Архитектура платформы

Платформа IACPaaS поддерживает три модели предоставления сервисов – PaaS, SaaS и DaaS. На рисунке 1 справа представлена модель предоставляемых сервисов платформы IACPaaS, слева – стандартные модели предоставления сервисов

Рис. 1. Различные модели предоставления сервисов

Fig. 1. Different models of service delivery

PaaS, SaaS и DaaS в облачных платформах. Рисунок позволяет сравнить набор компонентов платформы IACPaaS и наборы компонентов в других облачных платформах в соответствии с реализованными в их рамках моделями предоставления сервисов.

Архитектуру платформы можно разделить на четыре логических уровня: системный уровень, уровень управления, сервисный уровень и библиотечный уровень (рис. 2).

1. *Системный уровень* (уровень *Виртуальной машины*). Виртуальная машина платформы IACPaaS состоит из процессора информационных ресурсов, процессора решателей задач и процессора пользовательского интерфейса. Каждый из перечисленных процессоров предназначен для поддержки соответствующих архитектурных компонентов облачных сервисов.

2. *Библиотечный уровень* (уровень *Фонда платформы* и *Фонда пользователя*). Фонд платформы и Фонд пользователя – это множество единиц хранения различных видов и уровней абстрак-

ции (содержательно это хранимые базы данных и знаний, их концептуальные модели (метаинформация), решатели задач, агенты, шаблоны сообщений, модель абстрактного пользовательского интерфейса и т.п.).

3. *Сервисный уровень*. Данный уровень является совокупностью сервисов и оболочек интеллектуальных сервисов. Выделяются прикладные сервисы, создаваемые разработчиками для решения прикладных задач, а также оболочки и специализированные инструментальные сервисы, обеспечивающие поддержку соответствующих технологий разработки сервисов и оболочек, работ и необходимые для развития Фонда платформы.

4. *Уровень управления*. Данный уровень представлен базовым инструментальным сервисом *Система управления*, который предназначен для обеспечения контролируемого доступа к функциональным возможностям платформы IACPaaS и управления правами доступа на использование прикладных и инструментальных сервисов, разработанных на базе платформы.

Рис. 2. Концептуальная четырехуровневая архитектура платформы IACPaaS

Fig. 2. A conceptual four-level architecture of the IACPaaS platform

Заключение

В работе описаны основные характеристики и функциональные возможности облачной платформы IASaaS, которые отличают ее от других платформ.

Платформа поддерживает создание специализированных оболочек интеллектуальных систем. Их основная особенность и отличие от проблемно-независимых оболочек – ориентация на класс задач. С одной стороны, такие оболочки имеют более узкую по сравнению с проблемно-независимыми оболочками область применения, а с другой, они учитывают специфику предметной области и задачи. В связи с этим формирование баз знаний с использованием таких оболочек значительно упрощается, в процесс разработки могут быть включены эксперты предметной области, что соответствует современным требованиям к созданию интеллектуальных систем. Простота и понятность формирования баз знаний в таких типах оболочек, как отмечается в [8–10], значительно упрощают процесс разработки интеллектуальных систем и повышают их качество.

Платформа поддерживает создание оболочек ИС и сервисов по нескольким технологиям: базовой и специализированным. В настоящее время реализована поддержка двух специализированных технологий: *технологии разработки сервисов с интерактивными виртуальными средами и технологии создания сервисов с агентами, присоединенными к метаданной информации.* Каждая такая технология имеет свой набор инструментальных сервисов, упрощающих (по сравнению с базовой технологией) создание разрабатываемой оболочки или сервиса.

Платформа поддерживает создание новых специализированных технологий и работ, разработку необходимых для их поддержки инструментальных сервисов с использованием уже существующих технологий платформы.

Особенностью платформы является *поддержка запуска и выполнения сервисов* не только непосредственно на платформе, но и *на сторонних вычислительных платформах*, взаимодействие с приложениями которых возможно по протоколу http. Это обеспечивает гибкое создание оболочек и сервисов без ограничений на вычислительные мощности, используемые библиотеки, операционные системы и другие требования.

Решатели задач разрабатываются на основе *агентного подхода*, в соответствии с которым любой решатель – это множество агентов, обрабатывающих информационные ресурсы и обменивающихся между собой сообщениями.

Зарегистрированные пользователи платформы имеют личные кабинеты – *виртуальные рабочие места*, которые предоставляют им набор инструментов для управления данными и сервисами.

В настоящее время ведутся работы по расширению функциональных возможностей платформы, прежде всего по повышению уровня инструментальной поддержки различных технологий. Важными задачами являются усовершенствование методов и средств обеспечения безопасности платформы и защиты ее от некорректных действий пользователей, создание усовершенствованных методов единого программного интерфейса для доступа к единицам хранения Фонда и операций над информационными ресурсами.

Платформа поддерживает три модели облачных вычислений: PaaS, SaaS, DaaS.

Работа выполнена при частичной финансовой поддержке РФФИ (проекты №№ 16-07-00340 и 17-07-00299) и ПФИ ДВО РАН «Дальний Восток» (проект № 18-5-078).

Литература

1. Sun Le, Dong H., Khadeer Hussain F., Khadeer Hussain O., Chang E. Cloud service selection: State-of-the-art and future research directions. J. of Network and Computer Applications, 2014, vol. 45, pp. 134–150.
2. CloudServiceMarket. A comprehensive overview of Cloud Computing services. URL: <http://www.cloudservicemarket.info/default.aspx> (дата обращения: 06.10.2017).
3. Zhang Qi, Cheng Lu, Boutaba R. Cloud computing: state-of-the-art and research challenges. J. of Internet Services and Applications, 2010, no. 1, pp. 7–18. DOI 10.1007/s13174-010-0007-6.
4. Рыбина Г.В. Интеллектуальные системы: от А до Я. Серия монографий. В 3-х книгах. Книга 3. Проблемно-специализированные интеллектуальные системы. Инструментальные средства построения интеллектуальных систем. М.: Научтехлитиздат, 2015. 180 с.
5. Kumar S., Prasad R. Importance of expert system shell in development of expert system. Intern. J. of Innovative Research & Development, 2015, vol. 4, iss. 3, pp. 128–133.
6. Rajendra M. Sonar An enterprise intelligent system development and solution framework. Intern. J. of Applied Science, Engineering and Technology, 2007, vol. 4, no. 1, pp. 34–39.
7. Schalkoff R.J. Intelligent systems: principles, paradigms and pragmatics. Sudbury, Mass. Jones and Bartlett Publ., 2011, 758 p.
8. Артемьева И.Л. Специализированные оболочки интеллектуальных систем для сложно-структурированных предметных областей // 11-я Национ. конф. по искусств. интеллекту с междунар. участ. М.: ЛЕНАНД, 2008. Т. 1. С. 95–103.
9. Загоруйко Ю.А., Боровикова О.И. Программная оболочка для построения многоязычных тезаурусов предметных областей, ориентированная на экспертов // КИИ-2012: тр. 13-й Национ. конф. по искусств. интеллекту с междунар. участ. Белгород: Изд-во БГТУ, 2012. Т. 4. С. 76–83.
10. Загоруйко Ю.А. Семантическая технология разработки интеллектуальных систем, ориентированная на экспертов предметной области // Онтология проектирования. 2015. № 1. С. 30–46.
11. Musen M. The protégé project: a look back and a look forward. Newsletter AI Matters, 2015, vol. 1, iss. 4, pp. 4–12. DOI: 10.1145/2757001.2757003.
12. Осипов Г.С. Методы искусственного интеллекта. М.: Физматлит, 2015. 296 с.
13. Черняховская М.Ю. Формирование баз наблюдений на основе онтологии медицины // Информатика и системы управления. 2009. № 4. С. 198–200.
14. Князева М.А., Купневич О.А. Модель онтологии предметной области «Оптимизация последовательных программ». Определение расширения языка модели структурных программ

терминами потокового анализа // НТИ: Сер. 2. 2005. № 4. С. 14–22.

15. Артемьева И.Л., Рештаненко Н.В. Интеллектуальная система, основанная на многоуровневой онтологии химии // Программные продукты и системы. 2008. № 1. С. 84–87.

16. Грибова В.В., Федорищев Л.А. Визуализация виртуальных сред в облачных сервисах // Научная визуализация. 2016. Т. 8. № 1. С. 133–145.

17. Inzartsev A., Pavin A., Kleschev A., Gribova V., and Eliseenko G. Application of artificial intelligence techniques for fault diagnostics of autonomous underwater vehicles. Proc. OCEANS 2016 MTS/IEEE Conference & Exhibition, USA, Monterey, 2016. DOI: 10.1109/OCEANS.2016.7761098.

18. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. A two-level model of information units with complex structure that correspond to the questioning metaphor. Automatic Documentation and Mathematical Linguistics, 2015, vol. 49, no. 5, pp. 172–181.

19. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. Implementation of a model of a metainformation-controlled editor of information units with a complex structure. Automatic Documentation and Mathematical Linguistics, 2016, vol. 50, no. 1, pp. 14–25.

20. Rozenberg G. (Ed.) Handbook of graph grammars and computing by graph transformation, vol. 1: Foundations. World Scientific Publ. Co. Pte. Ltd., 1997, 572 p.

21. Ehrig H., Engels G., Kreowski H.-J., Rozenberg G. (Eds.). Handbook of Graph Grammars and Computing by Graph Transformation, vol. 2: Applications, Languages and Tools. World Scientific, 1999, 720 p.

22. Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Ч. 1: Разработка базы знаний и решателя задач // Программная инженерия. 2015. № 12. С. 3–11.

23. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. A Model for generation of directed graphs of information by the directed graph of metainformation for a two-level model of information units with a complex structure. Automatic Documentation and Mathematical Linguistics, 2015, vol. 49, no. 6, pp. 221–231.

24. Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Ч. 2: Разработка агентов и шаблонов сообщений // Программная инженерия. 2016. Т. 7. № 1. С. 14–20.

Software & Systems

DOI: 10.15827/0236-235X.123.527-536

Received 10.10.17

2018, vol. 31, no. 3, pp. 527–536

IACPaaS cloud platform for the development of intelligent service shells: current state and future evolution

V.V. Gribova¹, Dr.Sc. (Engineering), Senior Researcher, Deputy Director, gribova@iacp.dvo.ru

A.S. Kleshev¹, Dr.Sc. (Physics and Mathematics), Professor, Chief Researcher, kleshev@iacp.dvo.ru

F.M. Moskalenko¹, Ph.D. (Engineering), Senior Researcher, philipmm@iacp.dvo.ru

V.A. Timchenko¹, Ph.D. (Engineering), Senior Researcher, vadim@iacp.dvo.ru

L.A. Fedorichev¹, Ph.D. (Engineering), Research Associate, fleo@iacp.dvo.ru

E.A. Shalfeeva¹, Ph.D. (Engineering), Associate Professor, Senior Researcher, shalf@iacp.dvo.ru

¹ Institute of Automation and Control Processes Far Eastern Branch of RAS, Vladivostok, 690041, Russian Federation

Abstract. The paper describes the main features and functional capabilities of the IACPaaS cloud platform. It provides three models of cloud service delivery: PaaS, SaaS and DaaS. The platform is intended for development of specialized (i.e., oriented to specific domains and/or classes of solved problems) intelligent service shells, as well as for development of applied intelligent services using such shells. Intelligent service shells are also presented as cloud services of the platform. Development (maintenance) of applied services using a shell is reduced to formation (modification) of a knowledge base using the tools provided by this shell and binding it with a problem solver. The problem solver consists of a set of agents that are software components interacting with each other by exchanging messages.

Specialized intelligent service shells use a domain-specific conceptual representation, which is defined by the ontology of the domain for which the shell is created, for knowledge representation. The knowledge base formation tool uses such problem-oriented model (language) for knowledge representation to generate user interface oriented on domain experts. As a result, domain experts can form and maintain knowledge bases and databases within a familiar conceptual framework (without cognitive engineers as intermediaries or additional training) and make no mistakes in using the language for knowledge representation.

The IACPaaS platform provides a basic (universal) and a several specialized technologies for development of the applied intelligent service shells using tools that support these technologies. In addition, there is a technology of interaction between problem solver agents and external software (not included in the IACPaaS platform). It is based on the standard mechanism of HTTP request processing and the ability to run external software executable files from programs (scripts) located on a web server.

Keywords: expert system shells, intelligent systems, intelligent system development technology, cloud platform, cloud computing, intelligent services, multi-agent services.

Acknowledgements. The work has been partially financially supported by the RFBR (projects no. 16-07-00340, 17-07-00299) and PFR FEB RAS "Far East" (project no. 18-5-078).

References

1. Sun Le, Dong H., Khadeer Hussain F., Khadeer Hussain O., Chang E. Cloud service selection: State-of-the-art and future research directions. *J. of Network and Computer Applications*. 2014, vol. 45, pp. 134–150.
2. CloudServiceMarket. A Comprehensive Overview of Cloud Computing Services. Available at: <http://www.cloudservicemarket.info/default.aspx> (accessed October 6, 2017).
3. Zhang Qi, Cheng Lu, Boutaba R. Cloud computing: state-of-the-art and research challenges. *J. of Internet Services and Applications*. 2010, no. 1, pp. 7–18. DOI 10.1007/s13174-010-0007-6.
4. Rybina G.V. *Intelligent systems: from A to Z. Book 3. Problem-Specific Intelligent Systems. Instrumental Tools for Building Intelligent Systems*. Moscow, Nauchtekhizdat Publ., 2015, 180 p.
5. Kumar S., Prasad R. Importance of expert system shell in development of expert system. *Intern. J. of Innovative Research & Development*. 2015, vol. 4, iss. 3, pp. 128–133.
6. Rajendra M. Sonar an enterprise intelligent system development and solution framework. *Intern. J. of Applied Science, Engineering and Technology*. 2007, vol. 4, no. 1, pp. 34–39.
7. Schalkoff R.J. *Intelligent Systems: Principles, Paradigms and Pragmatics*. Sudbury, Mass. Jones and Bartlett Publ., 2011, 758 p.
8. Artemeva I.L. Specialized shells of intelligent systems for complex structured domains. *Proc. 11th National Conf. on Artificial Intelligence with Intern. Participation*. Moscow, LENAND Publ., 2008, vol. 1, pp. 95–103 (in Russ.).
9. Zagorulko Yu.A., Borovikova O.I. The software shell for building multilingual domain thesauri, focused on experts. *Proc. 13th National Conf. on Artificial Intelligence with Intern. Participation (KII-2012)*. Belgorod, BGTU Publ., 2012, vol. 4, pp. 76–83 (in Russ.).
10. Zagorulko Yu.A. A semantic technology for the development of intelligent systems, focused on domain experts. *Ontology of Designing*. 2015, no. 1, pp. 30–46 (in Russ.).
11. Musen M. The protégé project: a look back and a look forward. *Newsletter AI Matters*. 2015, vol. 1, iss. 4, pp. 4–12. DOI: 10.1145/2757001.2757003.
12. Osipov G.S. *Artificial Intelligence Methods*. Moscow, Fizmalit Publ., 2015, 296 p.
13. Chernyakhovskaya M.Yu. Formation of observation databases based on the medicine ontology. *Informatics and Management Systems*. 2009, no. 4, pp. 198–200 (in Russ.).
14. Knyazeva M.A., Kupnevich O.A. An ontology model for “optimizing sequential programs”. The definition of extensions for structural programs model language by terms of data flow analysis. *Scientific and Technical Information. Ser. 2. Information Processes and Systems*. 2005, no. 4, pp. 14–22 (in Russ.).
15. Artemeva I.L., Reshtanenko N.V. An intelligent system based on multilevel chemistry ontology. *Software & Systems*. 2008, no. 1, pp. 84–87 (in Russ.).
16. Gribova V.V., Fedorishchev L.A. Visualization of virtual environments in cloud services. *Scientific Visualisation*. 2016, vol. 8, no. 1, pp. 133–145 (in Russ.).
17. Inzartsev A., Pavin A., Kleshev A., Gribova V., Eliseenko G. Application of artificial intelligence techniques for fault diagnostics of autonomous underwater vehicles. *Proc. OCEANS 2016 MTS/IEEE Conf. & Exhibition*. Monterey, California, USA, 2016.
18. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. A two-level model of information units with complex structure that correspond to the questioning metaphor. *Automatic Documentation and Mathematical Linguistics*. 2015, vol. 49, no. 5, pp. 172–181.
19. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. Implementation of a model of a metainformation-controlled editor of information units with a complex structure. *Automatic Documentation and Mathematical Linguistics*. 2016, vol. 50, no. 1, pp. 14–25.
20. *Handbook of Graph Grammars and Computing by Graph Transformation. Vol. 1: Foundations*. G. Rozenberg (Ed.). World Sci. Publ. Co. Pte. Ltd., 1997, 572 p.
21. *Handbook of Graph Grammars and Computing by Graph Transformation. Vol. 2: Applications, Languages and Tools*. H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg (Eds.). World Scientific, 1999, 720 p.
22. Gribova V.V., Kleshchev A.S., Krylov D.A., Moskalenko F.M., Timchenko V.A., Shalfeeva E.A. A basic technology for developing intelligent services using IACPaaS cloud platform. Part 1. Development of a knowledge base and problem solver. *Software Engineering*. 2015, no. 12, pp. 3–11 (in Russ.).
23. Gribova V.V., Kleshchev A.S., Moskalenko F.M., Timchenko V.A. A Model for generation of directed graphs of information by the directed graph of metainformation for a two-level model of information units with a complex structure. *Automatic Documentation and Mathematical Linguistics*. 2015, vol. 49, no. 6, pp. 221–231.
24. Gribova V.V., Kleshchev A.S., Krylov D.A., Moskalenko F.M., Timchenko V.A., Shalfeeva E.A. A base technology for development of intelligent services with the use of IACPaaS cloud platform. Part 2. A development of agents and message templates. *Software Engineering*. 2016, vol. 7, no. 1, pp. 14–20 (in Russ.).