

СРАВНЕНИЕ СРЕДСТВ РАЗРАБОТКИ ЗНАНИЕ-ОРИЕНТИРОВАННЫХ СИСТЕМ ДЛЯ ИНТЕГРАЦИИ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ

Проведен сравнительный анализ технологий и стандартов построения распределенных систем с рассмотрением их возможностей для интеграции информационных ресурсов на основе онтологических описаний.

Ключевые слова: интеллектуальные системы автоматизированного проектирования и геоинформационные системы, онтологии, интеграция информационных ресурсов

Постановка проблемы

С развитием информационных технологий все большее распространение получили цифровые методы сбора, хранения, обработки и предоставления доступа к данным. Внедрение и использование таких методов привело к появлению множества распределенных источников данных, компьютерных моделей и приложений (далее – информационных ресурсов), находящихся в собственности государственных и частных организаций. Доступ к таким ресурсам осуществляется через Интернет и другие телекоммуникационные сети.

Форматы данных, которыми оперируют приложения, а также интерфейсы доступа к ним весьма разнообразны. Это зачастую приводит к явлению "лоскутной автоматизации" – ситуации, когда прямая интеграция информационных ресурсов между собой невозможна. Предвидеть все возможные применения ресурсов невозможно так как круг специалистов и пользователей слишком широк.

Проблема интеграции распределенных информационных ресурсов для решения широкого круга прикладных и исследовательских задач становится все более значимой и актуальной. Одним из примеров острой потребности в интеграционных усилиях могут быть информационные ресурсы для проектирования и управления такими сложными пространственно распределенными объектами, как городские инженерные и транспортные сети.

Анализ последних исследований и публикаций

В последнее время значительно возрос интерес к исследованиям в области создания знание-

ориентированных систем с использованием онтологий.

Онтология в компьютерных науках – это "спецификация концептуализации", т.е. явное описание представления человеком структуры определенной предметной области окружающего мира [1]. Формально онтология определяется как кортеж [2]:

$$O = \langle C, R, F \rangle,$$

где C – конечное множество концептов (понятий, терминов) предметной области;

R – конечное множество отношений между концептами; F – конечное множество функций интерпретации, заданных на концептах и/или отношениях онтологии O .

Одно из главных применений онтологий – информационная интеграция с учетом семантики понятий предметной области и ресурсов. Проблемы семантической интеграции данных и распределенных приложений рассмотрены в частности, в работах [3;4].

Вопросы описания и объединения онтологий при помощи специализированных языков и редакторов отражены во многих публикациях и не являются предметом исследования в этой статье.

В работе [5] рассмотрены преимущества и принципы применения онтологического подхода к построению распределенных систем для инженерных и транспортных сетей с целью интеграции приложений с функциональностью САПР, ГИС и автоматизированных систем управления (АСУ). В работе [6] представлены проблемы информационной безопасности использования описаний онтологий и средства их решения.

Формулировка цели статьи

Цель статьи – сравнительный анализ возможностей применения наиболее

распространенных технологий и стандартов к проектированию и реализации знание-ориентированных распределенных систем для инженерных и транспортных сетей, позволяющих интегрировать в себе приложения с функциональностью САПР, ГИС и АСУ.

Изложение основного материала

Выбор и обоснование критериев сравнения

Для сравнения средств разработки систем, обеспечивающих унифицированные методы доступа к распределенным информационным ресурсам на основе онтологий, используют пять критериев:

- наличие унифицированной среды взаимодействия приложений;
- гетерогенность интерфейсов доступа к ресурсам;
- каталогизация информационных ресурсов,
- встроенная поддержка онтологий;
- наличие реализаций с открытым исходным кодом.

Рассмотрим эти критерии более подробно.

Унифицированная среда взаимодействия приложений. Несмотря на то, что информационные сети гетерогенны по ряду характеристик (сетевые протоколы, операционные системы, форматы представления данных и методы доступа к ним), наличие гомогенной среды взаимодействия необходимо для обеспечения интероперабельности информационных ресурсов. Это обеспечивает возможность создания программных компонентов, работающих с любым ресурсом через такую среду по унифицированному протоколу.

Гетерогенность интерфейсов доступа к ресурсам. Для обеспечения взаимодействия приложений через гомогенную среду может потребоваться создание интерфейсов доступа к ним через гетерогенные информационные сети. Следовательно, технологии должны поддерживать работу с различными сетевыми протоколами и методами доступа к приложениям.

Каталогизация информационных ресурсов. Поиск ресурсов по заданным критериям должен осуществляться через каталог в том или ином виде. Централизация каталога необязательна.

Встроенная поддержка онтологий. Для обеспечения взаимодействия рассматриваемого класса знание-ориентированных систем необходимы средства сбора и хранения онтологий, а также выполнения запросов к хранилищам онтологий.

Наличие реализаций с открытым исходным кодом. Удовлетворение этому критерию позволит производить анализ и модификацию кода для совершенствования функционирования систем, а также выявления и устранения уязвимостей.

Выбор средств интеграции для сравнения.

Среди существующих технологий и стандартов, используемых для взаимодействия приложений, были выбраны следующие (потенциально наиболее подходящие для создания рассматриваемого класса знание-ориентированных распределенных систем):

- HLA (High Level Architecture) [7];
- Веб-службы на основе SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) и UDDI (Universal Description, Discovery and Integration) [8];
- стандарты FIPA (Foundation for Intelligent Physical Agents) [9].

Выбор этих объектов анализа в определенной степени отражает и эволюцию средств создания распределенных систем.

Сравнительный анализ и его результаты

Оценим рассматриваемые технологии и стандарты по выбранным критериям.

HLA – архитектура высокого уровня для создания распределенных имитационных приложений. Хотя архитектура разработана по заказу Министерства обороны США, она открыта для широкого сообщества разработчиков. Используя HLA, приложения могут взаимодействовать между собой вне зависимости от аппаратного и программного обеспечения, на основе которого они выполняются. Взаимодействие между приложениями регулируется посредством RTI (Run-Time Infrastructure) – инфраструктуры времени исполнения. Единицей модульности в HLA является федерат (federate). Обмен данными между федератами осуществляется при помощи взаимодействий (interactions).

Таким образом, информационные ресурсы могут быть представлены в виде федератов HLA и объединены в единую систему с унифицированным программным интерфейсом при помощи RTI.

В большинстве реализаций подключение федератов к RTI происходит напрямую через протоколы TCP/IP и UDP. Однако прямое соединение в реальном масштабе времени не всегда приемлемо [10], например, вследствие больших периодов ожидания при большой удаленности федерата от RTI. Полноценных реализаций с открытым исходным кодом, обеспечивающих подключение к RTI другими методами (например, посредством Веб-служб), на нет.

Каталог федератов не предусмотрен (предполагается, что федераты будут взаимодействовать с помощью сообщений, доставляемых подписанным на них федератам).

Встроенных средств поддержки онтологий в HLA нет.

Для HLA существует несколько реализаций с открытым исходным кодом. Одной из наиболее

популярных библиотек является Portico, реализованная на языке Java.

HLA полностью соответствует критерию наличия унифицированной среды взаимодействия приложений. Однако для полноценных разработок рассматриваемого класса распределенных систем потребуются усовершенствование одной из реализаций HLA в направлении поддержки онтологий и дополнительных методов доступа к RTI, а также создание каталогов ресурсов.

Веб-службы на основе стандартов Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Universal Description, Discovery and Integration (UDDI). Веб-службы – распространенный способ предоставления программного интерфейса через протокол HTTP, приобретающий со временем все большую популярность. Основными моделями программирования для создания Веб-служб являются REST и SOAP.

Веб-службы на основе REST сравнительно примитивны и используют базовые методы протокола HTTP (GET, POST, PUT, DELETE) для манипуляции данными.

Веб-службы на основе SOAP более функциональны и имеют средства для вызова Веб-методов (SOAP), описания интерфейсов вызовов этих методов (WSDL) и построения каталогов Веб-служб (UDDI).

SOAP – протокол обмена структурированными сообщениями в распределённой вычислительной среде. Посредством представления доступа к информационным ресурсам в виде Веб-служб на основе SOAP, достигается создание унифицированной среды взаимодействия ресурсов.

WSDL – язык описания Веб-служб, базирующийся на XML-синтаксисе. В WSDL-спецификации Веб-службы может содержаться информация, необходимая для правильных вызовов ее методов. UDDI – инструмент на синтаксической основе XML для каталогизации WSDL-описаний Веб-служб и последующего их поиска. В каталогах содержатся распределенные по категориям списки Веб-служб с общей информацией о них и ссылкой на WSDL-описание.

В основных стандартах Веб-служб онтологии не поддерживаются. Тем не менее, попытки интеграции онтологий с Веб-службами существуют. Примером может служить язык OWL-S, предназначенный для создания семантических Веб-служб и интеграции WSDL- и OWL-описаний, поддержки автоматического поиска и вызова таких Веб-служб, а также их объединения и взаимодействия с использованием имеющихся знаний. При определенных дополнительных усилиях, OWL-S может быть использован для

информационной интеграции Веб-служб на основе онтологий.

Вследствие широкого распространения стандартов Веб-служб, существует достаточно много их реализаций с открытым исходным кодом.

Благодаря своей универсальности и возможности запуска на большинстве современных аппаратных и программных платформ Веб-службы могут использоваться как среда взаимодействия приложений. Недостатком этого подхода является необходимость создания отдельной Веб-службы для каждого приложения, что может повлечь за собой дополнительные издержки (публичный Веб-сервер и Интернет-домен). Кроме того, Веб-службы предназначены для несвязанных между собой вызовов Веб-методов, в то время как активное взаимодействие некоторых типов приложений может потребовать интенсивного обмена сообщениями с сохранением промежуточных результатов.

Foundation for Intelligent Physical Agents (FIPA) – набор стандартов, образующих платформу для разработки интеллектуальных агентов и агентно-ориентированных систем в гетерогенных информационных средах.

Единицей модульности в FIPA является агент. Взаимодействие между агентами осуществляется с помощью сообщений на языке ACL (Agent Communication Language). Поиск агентов и знаний о них производится посредством службы каталогизации агентов DF (Directory Facilitator) и онтологической службы (Ontology Service).

Благодаря наличию онтологической службы, может быть полностью реализовано выполнение языковых запросов к хранилищам онтологий, что обеспечит информационную интеграцию агентов.

Для обеспечения доступа к информационным ресурсам в виде агентов осуществляется их интеграция в единую знание-ориентированную систему с унифицированной средой обмена данными. В соответствии со стандартами FIPA, доступ к такой среде также может производиться через гетерогенные среды, включая различные протоколы сетевого и прикладного уровней.

В стандартах FIPA определены также спецификации служб управления агентами AMS (Agent Management System) и политиками безопасности APSM (Agent Platform Security Manager), некоторых других полезных специализированных служб.

Стандарты FIPA довольно широко распространены. Наиболее популярной реализацией FIPA является платформа JADE [11], реализованная на языке Java и распространенная с открытым исходным кодом.

Таким образом, реализации стандартов FIPA в наибольшей степени удовлетворяют выбранным критериям. Это позволяет использовать их в качестве практической технологической основы для поэтапного создания систем взаимодействия информационных ресурсов инженерных и транспортных сетей.

Таблица 1

Результаты сравнения средств разработки

Критерий	Средства разработки		
	HLA	SOAP, WSDL, UDDI	FIPA
Унифицированная среда взаимодействия приложений	+	+/-	+
Гетерогенность интерфейсов доступа	-	+/-	+
Каталог информационных ресурсов	-	+	+
Встроенная поддержка онтологий	-	+/-	+
Наличие реализаций с открытым исходным кодом	+	+	+

Выводы

Проведено сравнение средств разработки распределенных систем для интеграции приложений с функциональностью САПР, ГИС и АСУ по выбранным пяти критериям.

Результаты сравнения показали, что все рассмотренные средства в той или иной степени применимы для разработки знание-ориентированных распределенных систем для инженерных и транспортных сетей, а также для решения других задач интеграции неоднородных информационных ресурсов.

На сегодня наиболее предпочтительным вариантом является использование распространяемых с открытым исходным кодом реализаций стандартов FIPA. Это позволит фактически без трудозатрат на усовершенствование платформы разрабатывать системы интеграции распределенных источников данных и приложений.

Выбор конкретного средства зависит исключительно от требований к разрабатываемой системе и предпочтений разработчиков, а также

является предметом дальнейших исследований в этом направлении.

Результаты сравнения средств разработки знание-ориентированных систем для интеграции распределенных приложений по выбранным критериям показаны в таблице.

Список литературы

1. Gruber T. A Translation Approach to Portable Ontology Specifications // Knowledge Acquisition Journal, vol. 5, 1993. – pp. 199-200.
2. Гаврилова Т.А., Базы знаний интеллектуальных систем /Т.А Гаврилова, В.Ф.Хорошевский . — СПб.: Питер, 2000. – 384 с.
3. Дишлик О. Інтеграція даних з використанням геоонтологій // Сучасні досягнення геодезичної науки та виробництва. – Львів, 2009. – № 1 (17). - С. 245-249.
4. Вовченко А.Е. Анализ и сравнение систем интеграции неоднородных информационных ресурсов/ Вовченко А.Е., Калининченко Л.А. // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 10 Всероссийской научной конференции "RCDL-2008". - Дубна: ОИЯИ, 2008. - С. 246-251.
5. Демченко В.В. Переваги онтологічного підходу до розподіленого моделювання інженерних і транспортних мереж /Демченко В.В. // Містобудування та територіальне планування: Наук. - техн. збірник. – К.: КНУБА, 2008. – Вип. 29. – С. 79-83.
6. Демченко В.В. Проблеми інформаційної безпеки використання описів онтологій прикладного рівня / Демченко В.В., Красовська Г.В. // Вісник Східноукраїнського нац. ун-ту ім. В. Даля. – Луганськ, СХУ ім. В. Даля, 2009.- № 6 (136), Ч. 1. – С. 24-27.
7. HLA Technical Specifications. WEB-caim / Спосіб доступу: <http://www.msco.mil/TechSpecs.html>
8. Web Services Architecture. WEB-caim / Спосіб доступу: <http://www.w3.org/TR/ws-arch/>
9. Foundation for Intelligent Physical Agents. WEB-caim / Спосіб доступу: <http://www.fipa.org/>
10. Möller B., Löfstrand B. Common federation performance bottlenecks. Proceedings of 2005 Fall Simulation Interoperability Workshop, 05F-SIW-044, Simulation Interoperability Standards Organization, Sweden, 2005. – p. 3-6.
11. JADE (Java Agent DEvelopment Framework). WEB-caim / Спосіб доступу: <http://jade.tilab.com/>

Статья поступила в редколлегию: 13.05.2010

Рецензент: д-р техн. наук, проф. В.М. Михайленко, Киевский национальный университет строительства и архитектуры, Киев.