

ИСПОЛЬЗОВАНИЕ SOA ДЛЯ РАЗРАБОТКИ РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ АСТРОСЕТИ

Постановка задачи

В последние десятилетия была осознана серьезная опасность, угрожающая человечеству, связанная с падением на Землю астероидов и комет. Стало очевидным, что удар тела размером в несколько километров крайне опасен не только для цивилизации современного вида, но и вообще для существования человечества (Gehrels, 1994; Remo, 1997). Поэтому, задача обнаружения малых тел Солнечной системы в рамках астероидно-кометного мониторинга околоземного пространства и выявления потенциально опасных небесных объектов приобретает большую актуальность.

Для решения этой задачи предлагается создать открытую сеть телескопов с малой апертурой, предназначенных для астероидного мониторинга околоземного космического пространства.

В состав астросети входят сеть телескопов-роботов, сеть пунктов обработки данных, центр управления и планирования.

Основная часть

Важным аспектом программы по созданию сети телескопов является разработка распределенной информационно-управляющей системы (РИУС), связывающей станции наблюдения с центрами обработки данных. Разработка и внедрение такой системы позволит оперативно анализировать получаемые результаты и гибко планировать и корректировать программу наблюдений.

Как правило, РИУС является развивающейся сложной гетерогенной иерархической интегрированной динамической системой с переменной структурой, нестационарными параметрами и нелинейными зависимостями характеристик от параметров и воздействий.

Таким системам, как объектам управления присущи следующие свойства:

Сложность. Сложная система определяется как система, которая не может быть осознана и изучена исследователем в целом.

Распределенность, как свойство системы, обеспечивает наилучшее согласование топологии системы управления с принципами организационно-технологического управления территориально и функционально распределённым объектом управления и исключает в системе циркуляцию избыточной информации при её параллельной и асинхронной обработке.

Структурная организация системы управления характеризуется широким использованием обратных связей, позволяющих проектировать развивающиеся системы управления удовлетворяющие требованиям устойчивости, управляемости (наблюдаемости), инвариантности, самоорганизации и т. д.

Рассматриваемый класс систем управления функционирует, как правило, в условиях неполноты и недостоверности информации о координатах и параметрах системы, неопределённости некоторых оценок и показателей. Это делает необходимым разработку системы управления в виде интеллектуальной системы, которая позволит динамически изменять свои параметры обеспечивая максимальную эффективность.

До недавнего времени попытки реализации таких систем были основаны на том, что компоненты системы представлялись как объекты, взаимодействующие посредством четко описанных интерфейсов. В качестве технологий построения в основном использовались два подхода: модели компонентных приложений Common Object Request Broker Architecture (CORBA) и Microsoft Distributed Component Object Model (DCOM).

Сравнительный анализ показал, что и CORBA и DCOM используют сильносвязанные взаимодействующие объекты, и внесение изменений в реализацию программных компонентов, предоставляющих и использующих определенный сервис, должны быть скоординированы между собой. Запросы к объектам (сервисам) в этих архитектурах, как правило, содержат небольшой объем информации, учитывающей специфику реализации сервисов ("мелкозернистая" – fine-grained – структура сервисов), и потому порождают значительный сетевой трафик между поставщиком и потребителем сервиса. В DCOM взаимодействие программных компонентов основано на закрытых интерфейсах Microsoft. CORBA не принадлежит частной компании, эта архитектура – плод усилий международного консорциума OMG, который ставил своей целью создание универсальной платформы интеграции разнородных программных компонентов на базе стандартного языка описания интерфейсов. Реализации спецификаций CORBA сильно варьируются в продуктах разных производителей, что ограничивает interoperability систем на базе CORBA. Кроме того, и CORBA и DCOM имеют существенные ограничения по поддержке действительно распределенных систем, их протоколы взаимодействия объектов слишком сложны для организации производительной связи сервисов, реализованных на разных машинах.

В настоящее время более перспективной технологий построения РИУС является сервис-ориентированная архитектура (SOA).

Разработка РИУС в соответствии с архитектурой SOA предполагает:

- использование явно независимых от реализации интерфейсов для определения сервисов;
- использование протоколов связи, усиливающих прозрачность расположения и функциональную совместимость;
- определение сервисов, инкапсулирующих многократно используемые бизнес-функции.

Как правило, набор сервисов реализуется посредством Web-технологии, однако это не является обязательным условием.

Для внедрения SOA необходимо напрямую или при помощи адаптеров создать интерфейсы сервисов для существующих или новых функций. Создание инфраструктуры на самом базовом уровне связано с предоставлением функций маршрутизации и доставки запросов сервисов нужному поставщику сервиса. Однако важно, чтобы инфраструктура поддерживала возможность замены одной реализации сервиса другой реализацией прозрачно для клиентов этого сервиса. Для этого необходимо не только определить интерфейс сервиса в соответствии с принципами SOA, но и обеспечить возможность вызова сервиса клиентским кодом способом, независимым от размещения сервиса и используемых протоколов связи. Описанные функции маршрутизации и замены в числе многих других функций предоставляет шина ESB.

Корпоративная сервисная шина Enterprise Service Bus (ESB) считается ключевым компонентом инфраструктуры SOA. ESB поддерживает взаимодействия с использованием сервисов, сообщений и событий в неоднородной среде, с надлежащим уровнем сервиса и управляемостью.

Среди множества функций, предоставляемых ESB, имеются функции: связи; взаимодействия сервисов; интеграции; обеспечения качества сервиса; обработки сообщений; управления сервисом.

Достоинство SOA состоит в том, что такая архитектура позволяет строить слабосвязанные приложения, то есть такие, в которых одну часть можно менять, не затрагивая другую. При этом SOA основывается на хорошо прописанных стандартах и сервисах, представленных так, что они могут быть обнаружены и задействованы конечным пользователем или приложением.

Использование SOA позволит унифицировать процесс доступа к телескопам и получения результатов наблюдения и сделать этот процесс независимым от используемого аппаратного и программного обеспечения на конкретном телескопе.

Таким образом, SOA предлагается использовать в качестве базовой технологии обеспечения сетевой инфраструктуры астронети.

В самом общем виде SOA предполагает наличие трех основных участников: поставщика сервиса, потребителя сервиса и реестра сервисов. Взаимодействие участников выглядит достаточно просто: поставщик сервиса регистрирует свои сервисы в реестре, а потребитель обращается к реестру с запросом.

Служба, таким образом, действует как программная компонента-посредник для приложения или ресурса. Такая функциональная парадигма дает пользователям этих приложений (или ресурсному пулу) возможность иметь дело исключительно с операционным описанием службы. Кроме того, в определении SOA подчеркивается, что все службы имеют адресуемые сетевые интерфейсы и взаимодействуют, используя стандартные протоколы и форматы данных, называемые сообщениями.

Взаимоотношение участников включает следующие основные аспекты:

- публикация сервиса;
- поиск сервиса;
- подключение и использование.

В соответствии с данной архитектурой и составом системы необходимо реализовать набор сервисов выполняющих следующие функции:

- формирование заявки на наблюдение;
- получение доступа к результатам наблюдения;
- планирование наблюдения на конкретном телескопе;
- динамическое перенаправление задачи наблюдения;
- аутентификация и авторизация пользователя;
- защита данных при передаче.

Список доступных сервисов публикуется на сайте центра управления системой. Для доступа к конкретному сервису пользователь должен пройти этапы аутентификации и авторизации. Данные функции могут быть основаны на проверке соответствия имени пользователя и пароля, которые задаются на этапе регистрации.

Для формирования заявки на наблюдение пользователь указывает координаты, время наблюдения и требования к срочности получения результатов (реальный масштаб времени или отложенная передача). Сервис планирования в соответствии с заданными параметрами, загрузкой телескопов и погодными условиями формирует задание на наблюдение для конкретного телескопа.

Задание на наблюдение высылается сервису управления телескопом. В процессе наблюдения, результаты накапливаются либо сразу передаются потребителю.

В случае выхода из строя телескопа или ухудшения погодных условий вызывается сервис динамического перенаправления задачи наблюдения. Данный сервис выполняет поиск свободных телескопов, удовлетворяющих поставленной задаче и направляет заявку на наблюдение одному из них.

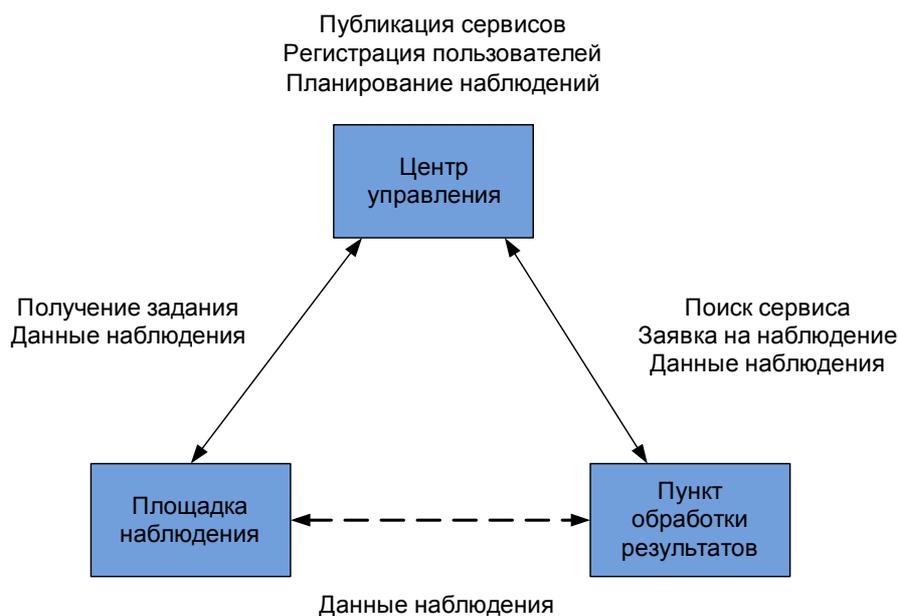
Каждый из сервисов представляет собой самодостаточную реализацию описанных функций с четко определенным интерфейсом, который устанавливает шаблоны для обмена сообщениями, используемые при взаимодействиях с функцией. Для описания и организации взаимодействия сервисов возможно использование следующих протоколов:

- eXtensible Markup Language (XML) – для представления данных;
- Web Services Definition Language (WSDL) – для описания доступных Web-сервисов;
- Universal Description, Discovery, Integration (UDDI) – для создания каталога доступных по сети Web-сервисов;
- Simple Object Access Protocol (SOAP) – для обмена данными.

Схема взаимодействия компонентов системы приведена на рисунке.

Исходя из того, что в качестве среды передачи будет использоваться Internet в качестве основных задач ESB, необходимо определить идентификацию сервиса, аутентификацию и авторизацию взаимодействующих сервисов и защиту передаваемых данных. Для решения задач защи-

ты передаваемых данных и возможно использование технологии виртуальных частных сетей (VPN).



При разработке РИУС астросети также необходимо учесть тот факт, что в системе циркулируют достаточно большие объемы данных. Например, объем одного снимка телескопа примерно равен 4 Мбт. Получая снимки с частотой 0,1 Гц, система генерирует трафик с интенсивностью 400 кб/с. Следовательно, эти требования необходимо учесть при формировании требований к пропускной способности каналов связи и дисциплин управления качеством в ESB.

В качестве технологии программной реализации возможно использование Net или J2EE. Эти технологии предоставляют инструментарий для разработки и тестирования сервисов, обеспечивают многократное использование создаваемых программных модулей и дают возможность представить существующие прикладные функции с помощью стандартных интерфейсов.

Заключение

Таким образом, использование SOA при разработке распределенной информационно-управляющей системы телескопов-роботов позволит унифицировать процесс доступа к результатам наблюдений и регистрации телескопов в качестве средств наблюдения. В сравнении с другими подходами (DCOM, CORBA) использование SOA значительно упрощает процесс расширения сети как площадками наблюдения (телескопами-роботами), так и пунктами обработки данных. Данный факт обусловлен тем, что SOA значительно снижает требования к программному обеспечению (как ОС, так и прикладному), используемому в точках предоставления и потребления сервиса.

Список литературы: 1. Норберт Биберштейн, Сенджей Боуз, Марк Фиаммант Компас в мире сервис-ориентированной архитектуры (SAO) КУДИЦ-Пресс, 2007. 256с. 2. Абдул Аллам, Андре Тост, Разработка Web-сервиса с использованием специализированного отраслевого стандарта обмена сообщениями, <http://www.ibm.com/developerworks/ru/library/ws-soa-messagingstandard/index.html> 3. Eric A. Marks, Michael Bell Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology Wiley, 2006. 384 p.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 12.11.2008