

УДК 004.415.2

Д.И. МАТЮШЕНКО, А.Б. ЛЕЩЕНКО

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ПОСТРОЕНИЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ НА БАЗЕ СЕРВИСА РАСПРЕДЕЛЕННОГО ДОСТУПА К ДАННЫМ ДЛЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Рассматривается способ построения мультиагентной системы на базе сервиса распределенного доступа к данным (DDS). Описаны преимущества использования DDS и рассмотрен способ реализации логики взаимодействия агентов в мультиагентной системе на основе определенных механизмов DDS.

агент, мультиагентные системы, распределенные системы, сервис распределенного доступа к данным (DDS), качество обслуживания (QoS)

Введение

Любая сложная мультиагентная система представляет собой информационную систему, базовыми задачами которой является обработка и передача данных. Одной из основных проблем решения задачи передачи данных является определение механизма взаимодействия элементов мультиагентной системы.

Постановка задачи. Анализ публикаций, проведенный по данной тематике, показал, что наработок в области использования распределенных систем реального времени в качестве транспортного механизма передачи данных в мультиагентных системах на базе сервиса распределенного доступа к данным (Data Distribution Service (DDS)) нет [1 – 5], поэтому работы по данному направлению являются актуальными.

В работе предлагается подход к построению сложных мультиагентных систем на основе DDS [6].

Метод решения

Понятия мультиагентных и распределенных систем. Исследовательские области мультиагентных систем и распределенных систем имеют очень много общего и формируют одну общую область исследования – распределенные вычисления на базе

распределенных агентов. Мультиагентные системы зачастую и являются распределенными системами, а распределенные системы в свою очередь являются базовой платформой для построения и поддержки мультиагентных систем. Это соотношение отображено на рис. 1 [7].



Рис. 1. Соотношение мультиагентных и распределенных систем

Агенты могут быть автономными (т.е. независимыми или неконтролируемыми), реактивными (т.е. обладающие определенной реакцией на определенные события), инициативными (т.е. самостоятельно инициатирующими определенные действия) а также коммуникативными (основной задачей которых является поддержка взаимодействия с другими агентами). Иногда к свойствам агента могут быть добавлены более строгие понятия (необходимые и желаемые требования, намерения). Агенты могут обладать различными возможностями; они могут быть статическими или мобильными, также могут быть

интеллектуальными и неинтеллектуальными. Каждый агент может обладать собственной задачей или ролью. Понятие агента и мультиагентной системы прежде всего используется как метафора для моделирования сложных распределенных процессов [8].

В свою очередь, принято считать, что распределенные системы представляют собой набор независимых систем, которые представляются конечному пользователю в виде единой целой системы.

Оба этих подхода (мультиагентный подход и подход распределенных систем) отвечают термину “распределенности”.

Область исследования мультиагентных систем направлена на решение распределенных задач, в свою очередь распределенные системы реализуются с целью решения проблем поддержки доступа к распределенной информации и ее обработки, а также для обработки распределенных процессов [9].

Исследовательская область распределенных вычислений на базе распределенных агентов является именно той областью, в которой мультиагентный подход и подход распределенных систем пересекаются. В понятиях этой исследовательской области можно утверждать, что мультиагентные системы являются определенным видом распределенных систем, где распределенные системы играют роль основы для реализации мультиагентных систем.

На данный момент существует целое множество самых разнообразных распределенных систем. Ко всем распределенным системам предъявляется одно общее требование – возможность передачи данных между различными исполняемыми потоками. Эти исполняемые потоки могут находиться как в пределах одного процесса, но также могут быть распределены между различными узлами сети. Не исключена возможность реализации такой схемы, когда используются оба варианта передачи данных между различными потоками (как в пределах одного процесса, так и между разными узлами).

Каждый из узлов сети или процессов могут быть

связаны посредством таких транспортных механизмов как Ethernet, shared memory, VME bus backplane, Infiniband. Базовые протоколы, такие как TCP/IP или протоколы более высокого уровня, такие как HTTP, могут обеспечивать взаимосвязь между различными узлами сети.

Одним из механизмов, который может быть использован для облегчения передачи данных между различными исполняемыми потоками, является сервис распределенного доступа к данным для систем реального времени, известный как DDS.

Сервис распределенного доступа к данным DDS. Сервис распределенного доступа к данным DDS является сетевым промежуточным программным обеспечением, которое значительно упрощает сетевое программирование. DDS использует модель “издатель/подписчик” (“publish/subscribe”) для рассылки и получения данных, событий и команд между различными узлами сети. Узлы сети, являющиеся источниками информации, (издатели) создают тематические блоки информации (“topics”) и публикуют ее в виде отдельных экземпляров (“instances/samples”). DDS берет на себя управление по доставке этих экземпляров информации ко всем конечным подписчикам, заинтересованным в данном тематическом блоке.

DDS выполняет все рутинные операции, связанные с передачей данных: адресация сообщений, маршрутирование и демаршрутирование данных (для получения возможности программной реализации “издателей” и “подписчиков” на различных платформах), доставка данных, контроль над потоками и т.д. Любой узел сети может быть как “подписчиком”, так и “издателем”, либо и тем и другим одновременно.

Модель DDS “издатель/подписчик” фактически устраняет сложность программирования распределенных приложений.

DDS поддерживает механизмы, которые выходят за рамки использования модели “подписчик/издатель”. Ключевым преимуществом является

тот факт, что приложения, использующие DDS для взаимодействия друг с другом, могут быть фактически не связаны. Совсем немного времени требуется для разработки способа их взаимодействия. В частности, имеется в виду то, что у приложения нет необходимости четко знать о других приложениях, участвующих в обмене информацией, включая даже информацию об их существовании и расположении. Как уже упоминалось выше, DDS берет на себя все заботы о доставке информации без непосредственного вмешательства со стороны пользовательского приложения, включая такие аспекты как:

- определение того приложения, которое должно получить сообщение;
- определение расположения получателя;
- определение реакции системы в том случае, если сообщение не было доставлено адресату.

Все это становится возможным благодаря тому, что DDS позволяет определять параметры “качества обслуживания” (Quality of Service (QoS)). Определение параметров QoS является важным и необходимым способом конфигурации механизмов автоматического обнаружения и определения поведения системы в процессе отсылки и получения сообщений. Эти механизмы могут быть настроены единожды без необходимости применения дополнительных усилий со стороны пользовательских приложений. Факт анонимного обмена сообщениями значительно упрощает разработку распределенных приложений и способствует созданию четко структурированных систем.

DDS также берет на себя решение проблемы “горячей замены” основных “издателей” запасными. “Подписчики” всегда могут рассчитывать на данные с высоким приоритетом доставки в том случае, если экземпляры этих данных все еще являются доступными (такowymi экземплярами данных могут считаться те, чей период достоверности все еще является актуальным). Система автоматически переключается к использованию основных “издателей” в случае их восстановления [6].

Преимущества использования DDS для построения мультиагентных систем. Любая сложная информационная система может быть рассмотрена как набор небольших автономных агентов, каждый из которых обладает своей функциональностью и своими свойствами, и четкое взаимодействие которых является необходимой составляющей для достижения общей целостности системы. Использование мультиагентных систем для построения сложных информационных систем обладает определенными преимуществами [7]:

- скорость и эффективность, благодаря использованию асинхронных и параллельных методов вычислений;
- простота и надежность, в смысле устойчивости системы к различного рода сбоям;
- гибкость и масштабируемость, благодаря простоте добавления новых агентов в систему;
- простота разработки и возможность повторного использования уже имеющихся компонент.

Все вышеперечисленные преимущества использования мультиагентных систем для построения сложных информационных систем являются четким отражением использования DDS как базовой распределенной системы для построения мультиагентной.

Несмотря на указанные преимущества, существует и ряд сложностей:

- декомпозиция основных задач, перераспределение подзадач и синтезирование полученных результатов;
- обработка распределенных данных, с целью сохранения целостности представления системы об окружающей среде;
- имплементация децентрализованного контроля и построения эффективного механизма координации действий агентов;
- разработка эффективных алгоритмов обучения мультиагентных систем;
- проблема представления данных;

– проблема коммуникации агентов, в смысле использования различных протоколов и реализации и использования агентов на различных платформах, а также в смысле использования различных способов коммуникации (каким образом и по какому принципу агенты должны взаимодействовать друг с другом);

- распределение ролей агентов в системе;
- обеспечение когерентности и стабильности поведения системы.

Значительная часть этих проблем может быть устранена (или сложность решения этих проблем может быть значительно уменьшена) при условии использования DDS в качестве базовой системы для построения мультиагентной системы.



Рис. 2. Взаимодействие агентов с DDS

На рис. 2 показан простейший пример, иллюстрирующий возможность использования DDS в качестве базовой транспортной системы для построения мультиагентной системы.

Реализация логики взаимодействия агентов.

DDS предоставляет различные механизмы, которые дают возможность реализовать логику взаимодействия агентов в мультиагентной системе и позволяют придать системам следующие свойства:

- скорость и эффективность, благодаря использованию асинхронных и параллельных методов вычислений;

- простота и надежность, в смысле устойчивости системы к различного рода сбоям;

- гибкость и масштабируемость, благодаря простоте добавления новых элементов в систему;

- простота разработки за счет повторного использования имеющихся компонент.

Одним из таких механизмов является сервис QoS.

Как уже было указано выше, определение параметров QoS является важным и необходимым способом конфигурации механизмов автоматического обнаружения и определения поведения системы в процессе отсылки и получения сообщений.

Сервис QoS обеспечивает “составляющую реального времени” распределенной системы DDS, так как определяет четкие правила взаимодействия элементов системы и передачи данных между ними.

Таким образом, уровень реализации логики взаимодействия агентов представляет собой ни что иное, как дополнительное DDS-приложение в мультиагентной системе (учитывая, разумеется, тот факт, что каждый агент в мультиагентной системе также является DDS-приложением).

Как показано на рис. 3, уровень реализации логики взаимодействия агентов состоит из следующих 4-х базовых модулей:

1. Ядро логики взаимодействия – непосредственный набор правил взаимодействия агентов.
2. Интерфейс взаимодействия с агентами.
3. Интерфейс определения правил взаимодействия агентов.
4. Интерфейс взаимодействия с DDS, основанный на определении параметров QoS.

Таким образом, основная задача уровня реализации логики взаимодействия агентов в мультиагентной системе сводится к трансформации правил поведения агентов, которые могут быть заданы исходя из задач мультиагентной системы, в правила пове-

дения элементов распределенной системы доступа к данным, в данном случае DDS.

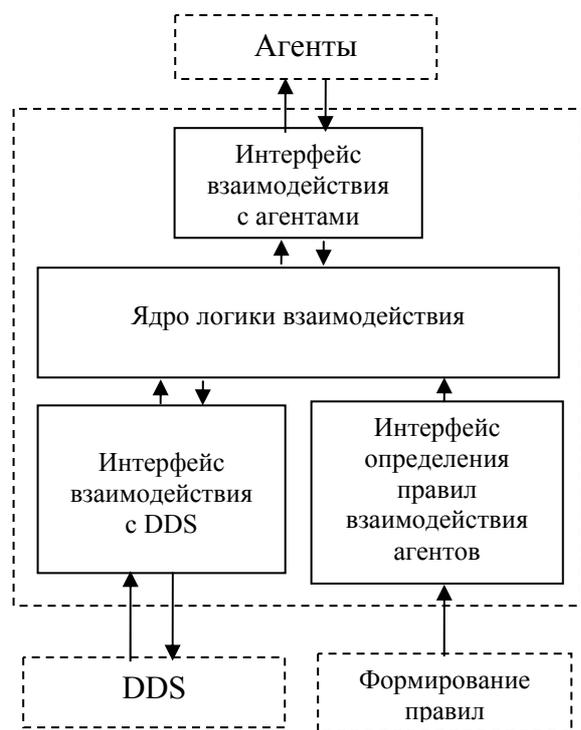


Рис. 3. Структура уровня реализации логики взаимодействия агентов

Выводы

Использование сервиса распределенного доступа к данным DDS открывает большие перспективы в области построения мультиагентных систем, и благодаря возможностям DDS позволяет решить ряд существующих проблем этой отрасли, а именно проблему общего протокола обмена данными между агентами в мультиагентной системе. Имеющиеся наработки в области построения распределенных приложений на базе сервиса распределенного доступа к данным DDS дают возможность использования этого сервиса в качестве транспортного уровня для построения мультиагентных систем, а также для реализации базовой логики взаимодействия агентов.

Литература

1. Protogerios N. Agent and WEB service technologies in virtual enterprises // Idea Group Inc (IGI), 2007. – 271 p.
2. Trajkovski G. Imitation-based approach to modeling homogenous agents societies // Idea Group Inc (IGI). – 307 p.
3. Lee J., Barley M. Intelligent agents and multi-agent systems: 6th Pacific Rim International Conference on Multi-Agents, Springer, 2003. – 215 p.
4. Sugumaran V. Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications // Idea Group Inc (IGI), 2007. – 2600 p.
5. Meersman R. On the move to meaningful internet systems 2005: OTM 2005 Workshops, Springer, 2005. – 1227 p.
6. Спецификация DDS [Электронный ресурс]. – Режим доступа: http://www.omg.org/technology/documents/dds_spec_catalog.htm (20.10.2007).
7. Zambonelli F., Omicini A. Challenges and Research Directions in Agent-Oriented Software Engineering // J. Autonomous Agents and MAS, 9. – 2004. – 146 p.
8. Luck M., McBurney P., Preist C. A manifesto for agent technology. Towards next generation computing // J. Autonomous Agents and MAS, 9. – 2004. – 252 p.
9. Amor M., Fuentes L., Vallecillo A. Bridging the gap between agent-oriented design and implementation using MDA, AOSE V, LNCS 3382. – 2005. – 108 p.

Поступила в редакцию 29.10.2007

Рецензент: д-р техн. наук, проф. В.М. Илюшко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.