

3. *Bollerslev T.* Generalized autoregressive conditional heteroskedasticity. //Journal of Econometrics, 31 (1986). С. 307–327.
4. *Holger Drees, Laurens De Haan and Sidney Resnick.* How to make a hill plot. //AMS 1991 subject classifications. Primary 60G70; secondary 60F15.
5. *McNeil A.* Estimating the tails of loss severity distributions using extreme value theory. //Astin Bulletin, 27(1997), С. 125–139.
6. *Mikosch T., Straumann D.* Whittle estimation in a heavy-tailed GARCH(1,1) model. //Stochastic processes and their applications, 100(2002), С. 187–222.

## **ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ МОБИЛЬНЫХ АГЕНТОВ**

**М. С. Шалай**

В компьютерном контексте под мобильным агентом понимают сущность, которая объединяет данные и код и способна перемещаться между разными средами выполнения [1]. Благодаря этому мобильный подход может дать ряд значительных преимуществ, таких как уменьшение передаваемого по сети трафика, децентрализация, высокая надежность и отказоустойчивость, а также легкость развертывания.

Все вышесказанное привело к тому, что парадигма мобильных агентов нашла свое применение в решении целого ряда серьезных и актуальных задач, самые значительные из которых: мобильные вычисления и различные задачи управления информацией, такие как поиск и обработка.

На данном этапе развития технологии нет стандартного и универсального подхода к программированию мобильных систем и агентов, что приводит к существенному различию в архитектуре и реализации систем. Последнее сильно влияет на интероперабельность – способность систем к взаимодействию друг с другом. Поэтому задача стандартизации стоит перед мобильной технологией особо остро. Учитывая уже существующие системы, как объектные, так и не объектные, были предложены аспекты стандартизации при разработке мобильной системы для обеспечения ее интероперабельности. К таким аспектам были отнесены: передача агентов и служебных классов или кода для необъектных систем между агентскими системами, а также управление агентами.

Распределенные вычисления в агентной системе осуществляются на основе перемещаемого агента (travelling agent). Большие объемы информации на сегодняшний день сильно затрудняют поиск и обработку данных. Использование агента на основе мобильных технологий, который странствует по сети в поисках нужной информации, позволяет уменьшить сетевой трафик за счет перемещения средств обработки данных к используемым ресурсам. Агент переходит с хоста на хост в поисках ин-

формации по заданным критериям, осуществляет ее отбор локально и возвращает лишь результаты.

При разработке и реализации мобильных систем приходится решать несколько сложных проблем: обеспечение переносимости и доступности кода, обеспечение атомарности выполнения и перемещения агентов, поддержка сетевого взаимодействия, обеспечение безопасности, многопоточная обработка, сериализации/десериализация объектов, загрузка классов [2].

Остановимся на таком важном вопросе как атомарность перемещения и безопасная сериализация агента. В мобильных системах на основе Java данная задача решается следующими способами:

1. Tryllian ADK. Активность агента разбита на отдельные задачи, каждая из задач возвращает статус завершения. Перемещения представляются задачей. Для перемещения агента сериализуется граф задач и текущее положение потока выполнения на этом графе.

2. DIET. Требование о перемещении агент передает своему упаковщику, который вызывает исключительную ситуацию, выполняет перемещения и заново запускает метод агента. Однако такой механизм не является полностью безопасным, например, исключение может быть перехвачено кодом агента, а не его упаковщиком, что повлияет на атомарность перемещения.

3. Kaariboga. Перемещения агентов отделены от логики, выполняемой агентом. Все перемещения агента специфицированы заранее: агент перемещается, новая среда запускает его и иницирует перемещение агента в соответствии с планом.

В разработанной системе реализуется механизм, похожий на способ DIET, но с некоторыми отличиями. Управление метода агента не прерывается при возникновении исключительной ситуации. Упаковщик агента помещает требование о перемещении в стек событий данного агента и ждет окончания выполнения метода агента, чтобы продолжить обработку сообщений. Когда упаковщик агента получает из стека событие с требованием о перемещении, он начинает сериализацию и перемещение. После перемещения обработка сообщений из стека продолжается. На рис. 1 представлен программный интерфейс разработанной мобильной системы.

Очевидно, что в случае мобильных агентов на каждом вычислительном узле должен существовать сервер, то есть платформа, предоставляющая функциональность для агентов. Эта среда должна быть безопас-

ной и учитывать такие свойства агентов как автономность, взаимодействие, мобильность.

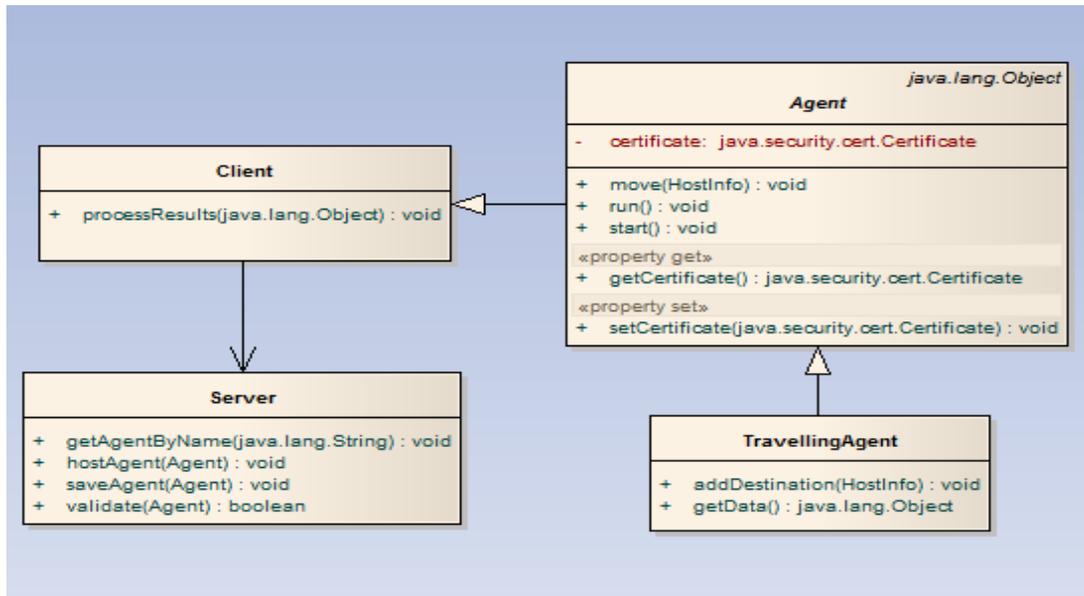


Рис. 1. Программный интерфейс мобильной системы

При рассмотрении парадигмы мобильных агентов может показаться, что она очень напоминает вирусы. Конечно, возможность скачать из сети, установить и выполнить неизвестный код создает угрозу безопасности. Таким образом, при разработке любой реальной системы необходимо глубоко разбираться в проблемах безопасности и создать исчерпывающую модель угроз [3]. Первая угроза состоит в том, что клиент может попытаться причинить вред хосту. В данном случае под клиентом понимается любая программа, которая взаимодействует с хостом через внешний интерфейс. Способ ослабления этой угрозы во многом зависит от коммуникационного протокола. Он не должен допускать анонимные вызовы и как минимум предусматривать аутентификацию агента. Защита коммуникационного протокола является очень важным аспектом, его нужно тщательно анализировать на этапе проектирования системы. Для распределенных систем реализован протокол, позволяющий аутентифицировать и шифровать запросы, что позволяет защитить хосты от атак многих типов. Что касается самого хоста, то одно хорошее правило разработки любых средств безопасности гласит, что "поверхность атаки" должна быть как можно меньшей, поэтому к выбору функциональности, предоставляемой внешнему миру, нужно подходить с большой осторожностью. На рис. 2 представлен вариант работы безопасной системы. Есть много способов разработки сетевых приложений. Мобильные агенты имеют свои области применения, достоинства и недостатками. Автоном-

ная и мобильная природа этих агентов помогает сократить сетевой трафик, обеспечивает децентрализацию, повышенную надежность и отказоустойчивость, а также легкость развертывания. А растущий интерес и область применения мобильных технологий бесспорно доказывает актуальность рассмотренной парадигмы.

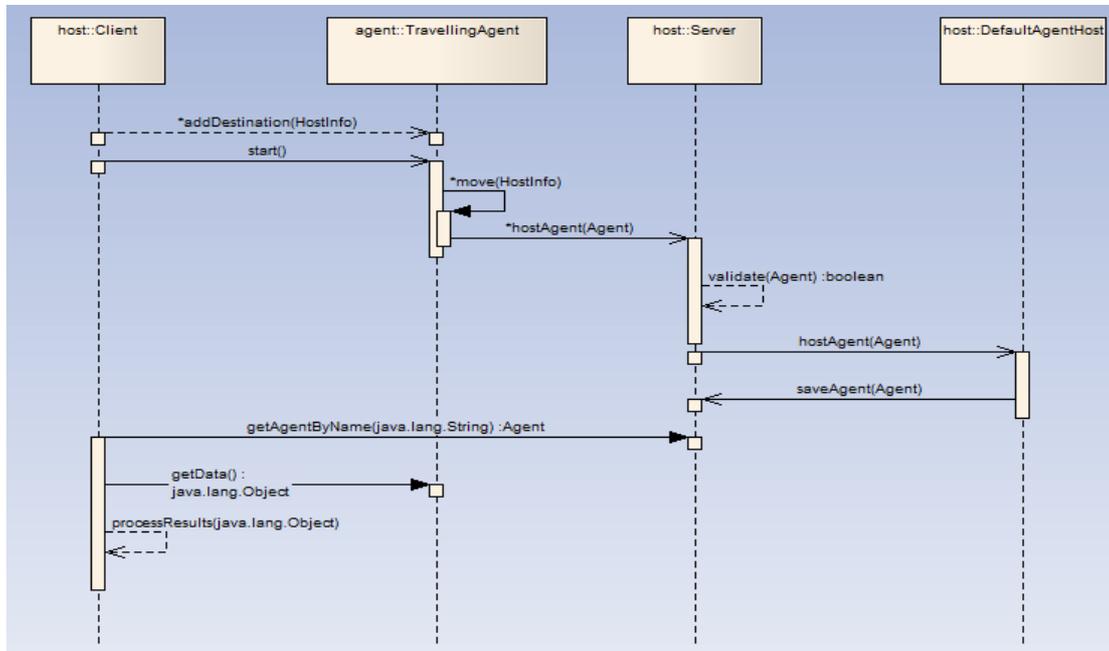


Рис. 2. Диаграмма работы безопасной системы

### Литература

1. *Danny B. Lange and Mitsuru Oshima.* Programming and Deploying JAVA Mobile Agents with Aglets. Addison-Wesley Longman Publishing, USA. – 1998, 256 p.
2. *Jason Weiss.* Java Cryptography Extensions – Practical Guide for Programmers. Morgan Kaufmann Publishers. – 2004.
3. *Tomas Sander and Christian F. Tschudin.* Protecting Mobile Agents Against Malicious Hosts, in Vigna, Giovanni (Ed.): Mobile Agents and Security, Springer-Verlag, 1998.