



Крунослав Тржец



Саша Дешич



Гордан Ежич



Марио Кушек

Крунослав Тржец, Саша Дешич

Эрикссон Никола Тесла а.о., Загреб, Хорватия
Ericsson Nikola Tesla d.d., Zagreb, Croatia

Гордан Ежич, Марио Кушек

Факультет электротехники и информатики
Университета в Загребе, Хорватия
Faculty of Electrical Engineering and Computing,
University of Zagreb, Zagreb, Croatia

Ключевые слова:

Агентская технология

Мобильные агенты

Многоагентская система

Дистанционное
управление

программным
обеспечением

Grid окружение

Key words:

Agent-oriented
technologies

Mobile agents

Multiagent system

Remote software
management

Grid environment

Дистанционное управление программным обеспечением с помощью агентов

Резюме

В данной статье описывается применение агентских технологий в области дистанционного управления программным обеспечением и информационно-коммуникационными системами. Особое внимание посвящено технологии мобильных агентов, которая использована при реализации дистанционной многоагентской системы управления программным обеспечением, с целью успешного удовлетворения запросов, связанных с гибкостью и автоматическим обслуживанием программного обеспечения в сетях новой генерации. Описан пример автоматизированного управления программным обеспечением в Grid окружении, в котором пользователи, программные компоненты и вычислительные ресурсы распределены вдоль большого количества сетевых узлов разнородных и географически рассредоточенных сетей связи. Многоагентская система дистанционного управления программным обеспечением создана в результате исследовательского проекта ROPE (*Remote Operation Environment* – Окружение с дистанционным управлением), осуществленного в сотрудничестве исследовательских отделов Института телекоммуникаций компании Эрикссон Никола Тесла и Факультета электротехники и информатики Университета в Загребе.

AGENT-BASED REMOTE SOFTWARE MANAGEMENT

Abstract

This article deals with usage of agent-oriented technologies in the area of remote software management in information communications systems. Special attention has been paid to mobile agent technology that was applied in implementation of multi-agent system for remote software man-

agement. The example of such an automated remote software management is given for the Grid environment that is characterized as an open system in which users, software components and computational resources are distributed across the heterogeneous and geographically dispersed telecommunications networks. The multi-agent system for remote software management is created as a result of the research project ROPE (Remote Operation Environment), which has been conducted by the Research Department at Ericsson Nikola Tesla's R&D Centre in cooperation with the Department of Telecommunications at the Faculty of Electrical Engineering and Computing, University of Zagreb.

1. Введение

Агентские технологии обеспечивают возможность развития систем, в которых программные агенты полностью автономным способом координируют свои действия при выполнении заданий, поставленных пользователями системы. Способность самостоятельного генерирования и осуществления целей на основании определенных заданий, является именно тем свойством, которое отличает автономных агентов от всех других типов программных компонентов (например, активных объектов). Чтобы агенты могли самостоятельно действовать и координировать свои действия, они должны обладать интеллектуальными свойствами, а в некоторых случаях и быть подвижными. Использование многоагентских систем, состоящих из интеллектуальных мобильных агентов, подтвердилось как очень эффективное решение для автоматизации деловых процессов поставщиков услуг и сетевых операторов в се-

тях телекоммуникаций [1].

Одним из таких деловых процессов является управление программным обеспечением в сетевых узлах (серверах), что включает дистанционное обслуживание и тестирование программных компонентов. Применение агентской технологии может полностью автоматизировать этот деловой процесс, что значительно снизит эксплуатационные расходы поставщика услуг и операторов сети, связанные с обслуживанием программного обеспечения. Именно поэтому, с целью реализации прототипа решения дистанционного управления программным обеспечением, был основан исследовательский проект ROPE. На этом проекте сотрудничали специалисты Института телекоммуникаций компании Эрикссон Никола Тесла и Факультета электротехники и информатики Университета в Загребе.

2. Агентские технологии

Программные агенты представляют сравнительно новую систему понятий в области программных инженерных изысканий, очень подходящую для решения проблемы мобильности, возникшую с появлением новой генерации сетей связи. Для этих сетей характерно объединение различных типов сетевых технологий и создание новых конвергентных услуг телекоммуникаций. Проблема мобильности услуги включает предоставление услуги мобильному пользователю, а также адаптацию услуги согласно требованиям пользователя и возможностям его оборудования. Из-за своих особенностей, многоагентские систе-

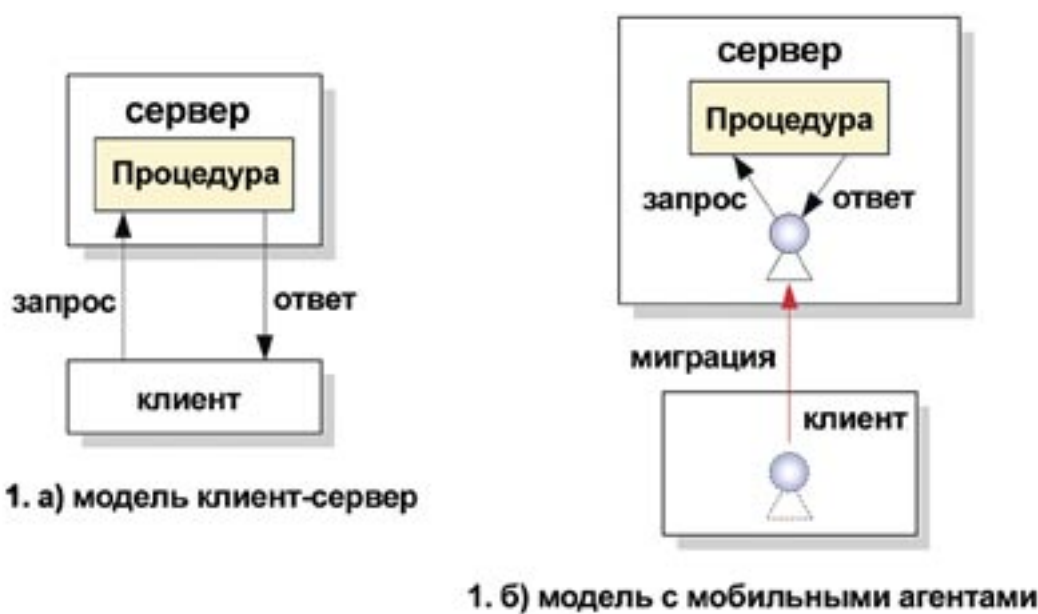


Рис. 1. Модели распределенного приложения

мы с мобильными агентами представляют эффективное решение для осуществления полной подвижности пользователя и услуг в новых условиях телекоммуникационных сетей.

Стремительное развитие программных агентов началось с использованием Интернет (IP) в информационно-коммуникационных системах [2]. До сегодняшнего дня в результате исследований агентских технологий разработаны различные агентские платформы, а также определены языки для коммуникации и протоколы для координации агентов [2, 3], что обеспечило возможность различных применений программных агентов в существующих больших распределенных системах (например, *Grid*), [4] и в (мобильных) телекоммуникационных сетях (например, UMTS) [5].

2.1. Свойства автономных программных агентов

Основной особенностью автономных программных агентов является самостоятельное решение задач, заданных им пользователями. Эта особенность автономных программных агентов отличает их от остальных программных компонентов (например, активных объектов). Ключевым элементом самостоятельного поведения агентов является их способность генерирования целей, результат которой может быть проактивное, т.е. упреждающее поведение. Самостоятельность программных агентов обеспечивает высокую степень автоматизации при решении заданий, а это особенно важно при выполнении большого числа сложных работ.

Автономные программные агенты могут отличаться в зависимости от степени интеллектуальности, которой они обладают, мобильности и возможностей взаимодействия. Интеллектуальность подразумевает способность учения и принятия решения на основании информации, которую агент получает из окружения. Интеллектуальность является основным свойством т.н. интеллектуальных агентов [6, 7]. Под мобильностью подразумевается способность передвижения агента в сети и представляет главную особенность т.н. мобильных агентов [8]. В отличие от остальных типов мобильных программных компонентов (например, мобильных объектов), которые не управляют своей миграцией, мобильные агенты могут полностью самостоятельно передвигаться по сети. Кроме того, мобильные агенты могут прервать исполнение своих задач и продолжить это на каком-то другом компьютере. Возможность взаимодействия позволяет агенту, в случае необходимости, осуществление коорди-

нированной коммуникации с другими агентами, с целью реализации совместной цели. Чтобы это было возможно, требуется определение языка коммуникации, который позволит обмен знаниями между агентами [2]. Сотрудничество является основной особенностью т.н. взаимодействующих агентов [9, 10]. Перечисленные свойства автономных агентов могут взаимно складываться, а в результате получают агенты с разными способностями.

2.2. Мобильные агенты

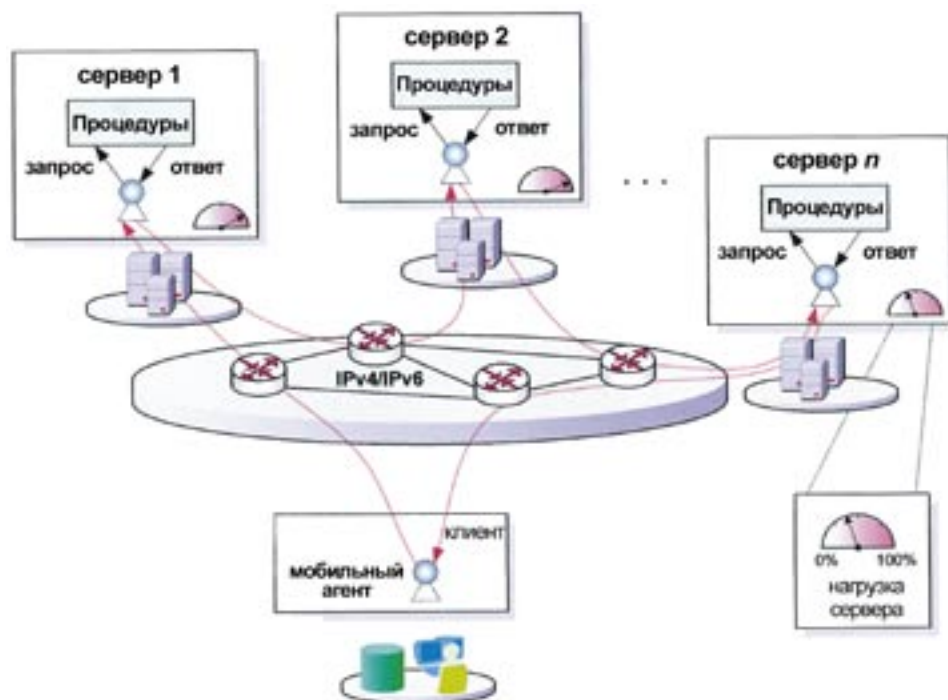
Программные агенты могут быть мобильными или стационарными. Стационарные программные агенты, в отличие от мобильных агентов, не обладают возможностью передвижения в системе. Поэтому они свои действия в многоагентской системе координируют исключительно с помощью агентских диалоговых протоколов [2]. Мобильные агенты могут использовать свою мобильность для эффективной координации своей деятельности, учитывая такие факторы как, например, загруженность сети, или выпадение из строя отдельных узлов в сети. Поэтому мобильные агенты особенно полезны для применения в системах связи.

В настоящее время большинство распределенных приложений основывается на модели клиент-сервер. Типичная реализация подразумевает взаимодействие посредством сети связи, в которой приложение клиента обменивается с сервером рядом запросов и ответов (Рис. 1. а). Использование технологии мобильных агентов позволило миграцию части распределенного приложения клиентов на компьютер сервера, и таким образом удаленное взаимодействие заменено локальным (Рис. 1. б).

Из перспективы пользователя распределенного приложения, модель с мобильными агентами имеет несколько преимуществ по сравнению с моделью клиент-сервер. Во-первых, при использовании мобильных агентов можно значительно уменьшить загруженность телекоммуникационной сети в случае, если задачи, которые нужно выполнить в удаленных узлах, могут вызвать интенсивную коммуникацию между клиентом и сервером.

Во избежание обмена большим количеством информации между сетевыми узлами, мобильный агент посылается посредством IP сети в удаленный сетевой узел (сервер) и там выполняет обработку данных, а после окончания обработки возвращается с результатами. Во-вторых, мобильные агенты также можно применять при реше-

Рис. 2. Миграция мобильного агента в IP сети



нии проблемы ограниченности вычислительных ресурсов, например, мигрируя в IP сети и самостоятельно находя сервер, располагающий достаточным количеством вычислительных ресурсов, мобильный агент может полностью автоматически выполнить задачу, заданную ему пользователем распределенного приложения (Рис. 2.).

2.3. Агентская платформа

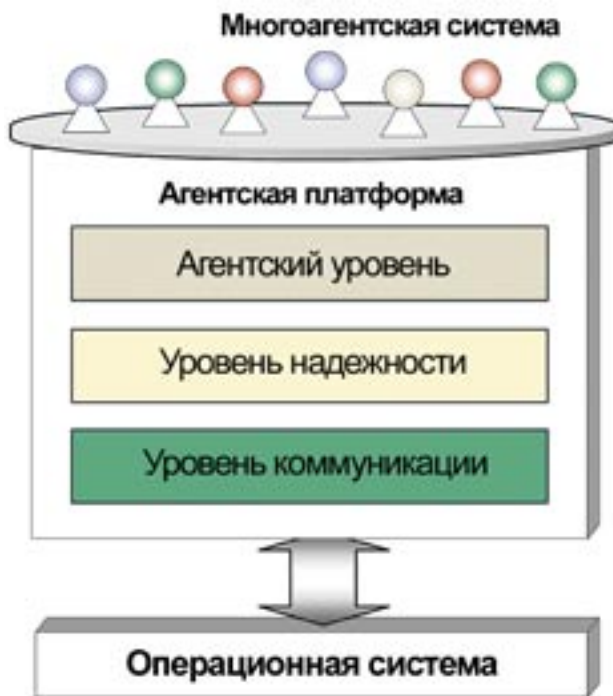
Чтобы агенты обладали возможностью доступа и действия в каком-то узле сети, должно существовать соответствующее окружение (платформа), которое может принять мобильного агента и обеспечить ему выполнение задания. Окружение или платформа управляет жизненным циклом агентов, т.е. их созданием и уничтожением, а также обеспечивает их миграцию и коммуникацию [6]. Агентская платформа (агентство) находится в узле сети (сервере). Группа связанных агентских платформ образует агентское окружение, внутри которого агенты перемещаются и выполняют операции. Типичная агентская платформа предоставляет следующие механизмы:

- управление агентами;
- надежность (конфиденциальность и целостность информации);
- обеспечение коммуникации между агентами.

Агентская платформа позволяет реализацию многоагентской системы, которая представляет распределенное приложение, основывающееся

на агентах. Большинство платформ имеет архитектуру, состоящую из трех уровней – агентского уровня, уровня надежности и уровня коммуникации (Рис. 3.). Агентский уровень служит для

Рис. 3. Пример архитектуры агентской платформы



управления агентами (например, их создания, передвижения, идентификации, и т.д.). Уровень надежности обеспечивает механизмы для защиты тайности информации и защиты от неуполномоченных действий в информационно-коммуникационной системе. Механизмы надежности нужны для осуществления надежной коммуникации и миграции агентов. Уровень коммуникации служит для доставки сообщений, которыми обмениваются агенты.

2.4. Преимущества мобильных программных агентов

Применение мобильных агентов предоставляет целый ряд преимуществ при развитии распределенных приложений [8]. Ниже перечислены преимущества мобильных агентов, которые были ключевыми при выборе этой технологии для развития распределенных приложений (многоагентской системы), предназначенных для дистанционного управления программным обеспечением:

- меньшая загруженность сети (экономится ширина полосы передачи);
- самостоятельное выполнение задач (автономность);
- возможность работы распределенного приложения без постоянной связи с сетью (автономный режим работы - *offline*);
- уменьшение запросов на использование клиентских ресурсов (т.е. меньшая загруженность клиента);
- конкурентное выполнение (реализовано посредством многоагентской системы).

Меньшая загруженность сети достигается уменьшением числа размещаемых в сети сообщений. Т.к. мобильный агент перемещается на удаленный сервер, и вся коммуникация выполняется локально, нет потребности в постоянной коммуникации между клиентом и сервером. Режим работы без потребности постоянной связи посредством сети можно осуществить благодаря мобильности агентов. Агент посылается серверу и возвращается клиенту после самостоятельно выполненной работы. Самостоятельность проявляется в том, что мобильный агент выполняет задание без дополнительной интеракции с клиентом. Мобильные агенты уменьшают запросы в отношении ресурсов на стороне клиента, т.к. большинство действий выполняют на стороне сервера. При исполнении более сложных работ, отдельные части которых конкурентные, можно создать и обеспечить работу большему числу агентов, которые в таком случае задания выполняют параллельно. Такое параллельное выполнение заданий

ведет к уменьшению времени выполнения совокупной работы, а также к оптимальному использованию процессорных емкостей компьютеров в сети. Кроме того, есть возможность перемещения объемной работы со слабого на более мощный компьютер и, таким образом, уменьшения вероятности перегрузки в отдельных узлах сети.

3. Дистанционное управление программным обеспечением

Дистанционное управление программным обеспечением в распределенных системах очень взыскательная работа, особенно если в такой системе есть большое число узлов (компьютеров). Администраторы таких систем должны выполнить сотни операций, связанных с управлением программным обеспечением, таких как, инсталляция, запуск или завершение программных компонентов. Кроме того, в сетях с мобильными пользователями и терминалами обслуживание программного обеспечения дополнительно усложняется.

Одна из больших проблем управления программным обеспечением в телекоммуникационных сетях новой генерации связана с тестированием новых программных компонентов, т.к. результаты тестирования программного обеспечения в контролируемом лабораторном окружении могут отличаться от результатов в реальном окружении на удаленном узле. Поэтому требуется выполнение и дистанционных тестирований в узле, в котором этот новый программный компонент будет установлен. Эту проблему можно эффективно решить с помощью мобильных агентов [11], а особенно если существует потребность тестирования программных компонентов на большем числе узлов телекоммуникационной сети [12].

3.1. Управление программным обеспечением в сети связи

Конвергенция услуг и сетевой инфраструктуры в новой генерации телекоммуникационных сетей все чаще вызывает у поставщиков услуг и операторов сети потребность в эффективной автоматизации процесса управления программным обеспечением в различных типах сетевых узлов. Разнородность сетевой инфраструктуры, терминалов и услуг, а также мобильность пользователей представляет новый вызов, с которым сталкиваются администраторы систем управления и надзора сетью и услугами. С целью управления программными компонентами, установленными в различных узлах (серверах) сети, администра-



Рис. 4. Концепт работы системы RMS

торы применяют различные приложения для дистанционного обслуживания программного обеспечения. Однако большинство этих (распределенных) приложений обеспечивает ограниченную степень автоматизации процесса дистанционного управления программным обеспечением, а особенно если это управление относится на большое число (взаимно зависимых) программных компонентов, рассредоточенных на большом числе сетевых узлов, как в случае новой генерации сетей связи.

Чтобы обеспечить возможность обслуживания программных компонентов в сетях новой генерации, исследователи Института телекоммуникаций компании Эрикссон Никола Тесла и Факультета электротехники и информатики Университета в Загребе разработали систему дистанционного управления программным обеспечением (RMS¹ - Remote Maintenance Shell), которая основывается на применении технологии мобильных агентов. Система RMS обеспечивает полностью автоматизированное дистанционное управление программным обеспечением на большом числе узлов в IP сети [13].

3.2. Система дистанционного управления программным обеспечением, RMS

Пользователю системы дистанционного управления программным обеспечением, RMS, дается возможность простого и эффективного управления программным обеспечением, применяя многоагентскую систему на агентской платформе

Grasshopper [14]. Система содержит мобильные агенты, служащие для выполнения различных операций, связанных с дистанционным обслуживанием и тестированием программных компонентов. Управление полностью автоматизировано и пользователь лишь должен специфицировать желаемую RMS услугу, связанную с управлением определенным программным компонентом (например, инсталляция, замена версии, и т.д.), а система RMS реализует услугу, учитывая загруженность сети и мобильность пользователя в сети связи.

Система RMS состоит из управляющей станции, размещенной в узле (компьютере) управления, и из обслуживаемого окружения, расположенного в удаленных узлах сети (Рис. 4.), [11]. На управляющей станции находится консоль (пульт оператора) с агентами, которые посредством графического интерфейса накапливают пользовательские запросы и представляют результаты выполненных операций обслуживания и тестирования. Окружение для обслуживания обеспечивает управление программными компонентами на определенном удаленном узле сети.

Каждый программный компонент, которым намереваемся дистанционно управлять и тестировать с помощью системы RMS, должен быть разработан согласно определенным правилам. Это приспособление программного компонента к системе RMS выполнено посредством особого слоя адаптации, т.н. испытательного окружения. Такой подход обеспечивает стандартизованный интерфейс между системой и программными компонентами.



Рис. 5. Параллельная работа версий программного компонента

Посредством графического интерфейса пользователь специфицирует желаемые операции обслуживания или тестирования над определенными программными компонентами на удаленных узлах, затем агенты в управляющей станции составляют план выполнения желаемых операций и создают мобильные многооперационные агенты, которым выделяют соответствующие задачи, установленные планом управления. Для выполнения этих задач каждый мобильный агент перемещается на соответствующий удаленный узел в информационно-коммуникационной системе (Рис. 4).

Система RMS предлагает пользователю основные и передовые возможности дистанционного управления программным обеспечением [11]. Основные услуги системы RMS содержат дистанционную инсталляцию и управление версиями программных компонентов, при чем мобильные агенты системы RMS ответственны:

- за миграцию определенной версии программного компонента и испытательного окружения из управляющей станции в удаленные узлы;
- за инсталляцию, запуск, завершение и деинсталляцию соответствующей версии программного компонента в удаленных узлах.

Передовые возможности системы RMS включа-

ют возможности дистанционного тестирования и надзора работы различных версий программных компонентов в удаленном узле. Для потребностей тестирования система RMS может поддерживать одновременную работу нескольких версий одного и того же программного компонента. Возможны три различных способа работы версий программного компонента:

- обычная работа, когда в каждом отдельном моменте активна только одна версия управляемого программного компонента;
- параллельная работа (Рис. 5.);
- выборочная работа (Рис. 6.).

3.2.1 Параллельная работа

Поддерживая одновременное выполнение двух версий (старой и новой) определенного программного компонента в удаленном узле, режим параллельной работы позволяет дистанционное тестирование новой версии программного компонента, сравнивая выходные данные новой и старой версий. При проверке работы новой версии программного компонента сравнение особенно эффективно на тех частях, которые не менялись в отношении на старую версию. Таким образом, возможно непосредственно, в удаленном узле,



Рис. 6. Выборочная работа версий программного компонента

установить совместимость новой и старой версий программного компонента.

При параллельной работе программного компонента одинаковые входящие данные посылаются в обе версии, а затем сравниваются полученные выходные данные. Если выходные данные старой и новой версии различные, тогда выходные данные старой версии воспринимаются как релевантные, а новая версия гасится из-за невозможности синхронизации со старой версией, которая продолжает нормально работать и обрабатывать запросы.

3.2.2 Выборочная работа

После какого-то периода успешной параллельной работы программный компонент переходит в фазу выборочной работы. В этом режиме работы обе версии программного компонента работают таким образом, что часть запросов обрабатывает новая версия, а остаток старая версия. Постепенно увеличивая число запросов, направленных новой версии, обеспечивается постепенная замена старой версии программного компонента новой версией.

В отличие от параллельного режима работы программного компонента, в котором старая и новая версии обрабатывают одни и те же запросы, в выборочном режиме работы старая и новая версии обрабатывают различные запросы, т.е. их выходные данные различные. Поэтому система RMS обеспечивает доставку выходных данных и старой, и новой версии программного компонента.

4. Применение системы RMS в Grid окружении

Прототип системы RMS испытан в процессе управления программной системой под названием MonALISA (*Monitoring Agents in a Large Integrated Services Architecture* - Контрольные агенты в обширной архитектуре интегрированных услуг), [15] которая представляет распределенное приложение в *Grid* окружении Европейского исследовательского центра CERN. Для *Grid* окружения характерно распределение пользователей, программных компонентов и вычислительных ресурсов между большим числом сетевых узлов, находящихся в разнородных и географически рассредоточенных телекоммуникационных сетях [16]. Первоначальные результаты исследовательского центра CERN при тестировании прототипа RMS в *Grid* окружении показали, что применение технологии мобильных агентов очень полезно

для автоматизированного управления программным обеспечением в информационно-коммуникационных системах с большим количеством узлов, т.к. существует потребность в одновременном (параллельном) выполнении операций в нескольких узлах [10].

4.1. Программная система MonALISA

Программная система MonALISA [15] предназначена для распределенного надзора работы услуг в *Grid* окружении с целью обеспечения улучшенного использования ресурсов в такой большой распределенной системе. Система MonALISA обеспечивает возможность интеграции существующих приложений для отслеживания и надзора распределенных систем, а также осуществление процедур накопления данных о сетевых узлах и программных компонентах.

Так как система MonALISA используется на большом числе узлов в *Grid* окружении CERN, обслуживание и наращивание такой программной системы, содержащей большое число взаимно зависимых программных компонентов, требует больших усилий администраторов информационно-коммуникационной системы исследовательского центра CERN.

Существующий механизм управления программным обеспечением в приложении MonALISA периодически проверяет, существует ли новая версия программного компонента. Если обнаружит существование новой версии определенного программного компонента, инициируется процесс наращивания/модернизации, который заканчивается запуском новой версии программы. Такой способ замены версий программных компонентов предоставляет лишь частичную автоматизацию процесса управления программным обеспечением в *Grid* окружении. Поэтому применена система RMS с целью введения следующих улучшений:

- инициация процесса наращивания программных компонентов во время наименьшей загрузки ресурсов *Grid* окружения;
- постепенное введение новых версий программных компонентов;
- автоматическое возвращение старой версии программного компонента, если ново установленная версия не удовлетворительна.

4.2. Сценарий замены версии программного компонента

Применение системы RMS, базирующейся на технологии мобильных агентов, обеспечивает полную автоматизацию процессов инсталляции и запуска определенного программного компонента приложения MonALISA в *Grid* окружении. Администратор информационно-коммуникационной системы должен лишь определить, на каких сетевых узлах желает замену существующей версии определенного программного компонента новой версией. В продолжение статьи рассматривается сценарий замены версии программного компонента в большем числе узлов в *Grid* окружении [17].

Программный компонент устанавливается на удаленном узле с помощью скрипта инсталля-

ции, написанного декларативным языком ANT [18], который позволяет спецификацию очень сложных процедур инсталляции для различных операционных систем. Все эти процедуры, специфицированные администратором информационно-коммуникационной системы, запускаются в удаленном узле с помощью мобильных агентов системы RMS. После того как администратор системы, посредством графического интерфейса в управляющей станции, определит узлы, в которых требуется замена определенного программного компонента, создаются RMS мобильные агенты для выполнения требуемой замены версии. Мобильный агент может мигрировать в удаленный узел и новую версию программного компонента, и в скрипт инсталляции этой версии (Рис. 7), или же может мигрировать только в скрипт инсталляции, а новую версию программного компонента получить с *Web* сервера (Рис. 8).

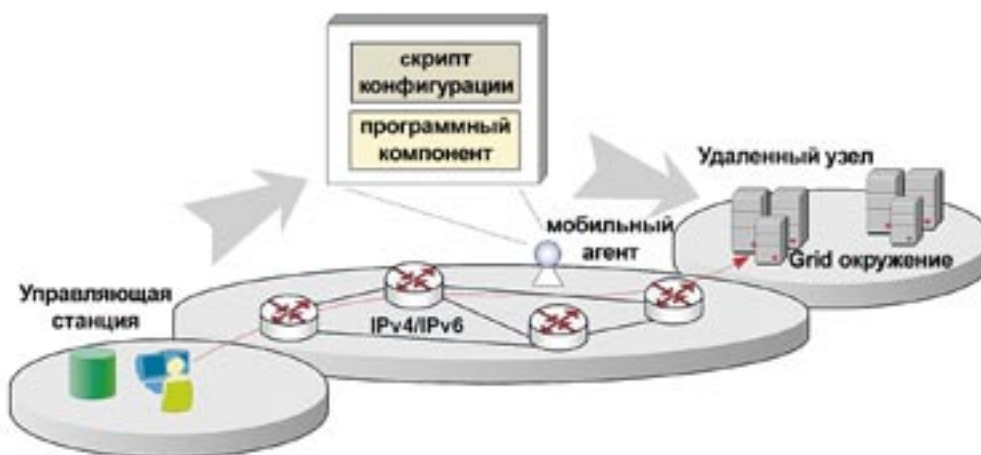


Рис. 7. Перемещение программного компонента с помощью мобильного агента

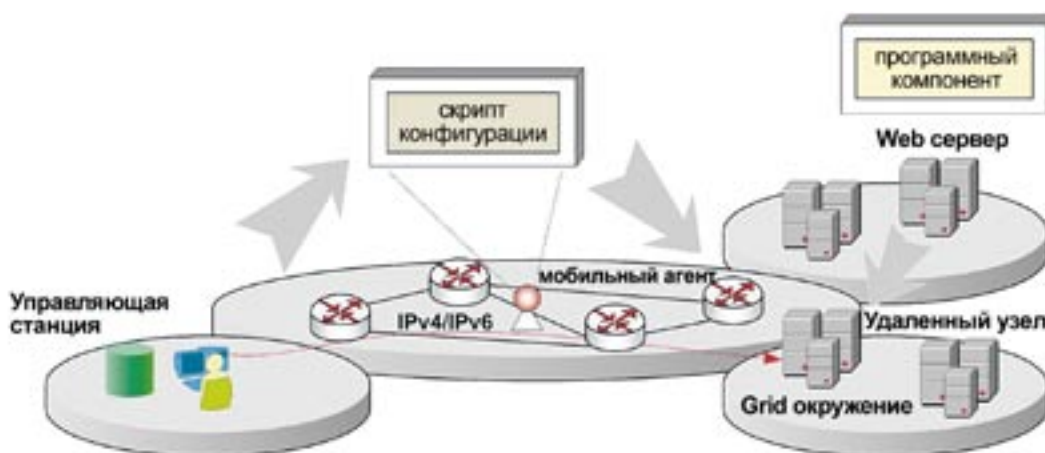


Рис. 8. Перемещение программного компонента посредством Web сервера

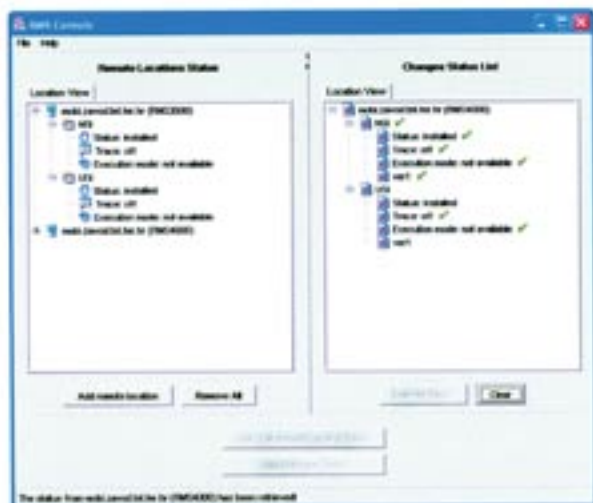


Рис. 9. Графический интерфейс системы RMS

Администратор информационно-коммуникационной системы может наблюдать за процессом замены программного компонента на определенном удаленном узле посредством графического интерфейса в управляющей станции (Рис. 9). Графический интерфейс системы RMS очень наглядным способом разрешает просмотр всех изменений, выполненных на определенном программном компоненте.

5. Вывод

Применение агентских технологий обеспечивает полную автоматизацию процесса дистанционного управления программным обеспечением в информационно-коммуникационных системах с большим числом сетевых узлов, например, в *Grid* системах, снижая расходы по их обслуживанию. Кроме того, применение мобильных агентов эффективно решает целый ряд проблем, связанных с дистанционным обслуживанием и тестированием программных компонентов в новой генерации сетей связи, для которых характерна разнородность сетевой инфраструктуры, терминалов и услуг, а также мобильность пользователей.

Поставщики услуг и операторы сетей, используя систему RMS, могут эффективно дистанционно управлять программным обеспечением. Применение мобильных агентов в системе RMS обеспечивает возможность координации активностей, связанных с дистанционным параллельным обслуживанием нескольких узлов с тестированием программных компонентов, учитывая проблемы нагрузки и подвижности пользователей в сети связи.

Список сокращений

Grid	<i>Grid</i> открытая система, в которой пользователи, программные компоненты и вычислительные ресурсы распределены вдоль большого количества сетевых узлов разнородных и географически рассредоточенных сетей связи
IP	<i>Internet Protocol</i> Протокол Интернет
MonALISA	<i>Monitoring Agents in a Large Integrated Services Architecture</i> Контрольные агенты в обширной архитектуре интегрированных услуг
RMS	<i>Remote Maintenance Shell</i> Система дистанционного обслуживания
ROPE	<i>Remote Operations Environment</i> Окружение с дистанционным управлением
UMTS	<i>Universal Mobile Telecommunications System</i> Универсальная система мобильной связи

Литература

- [1] Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sessler, R., Albayrak, S., "Agent-based Telematic Services and Telecom Applications", *Communications of the ACM*, vol. 44, no. 4., pp. 43-48, 2001.
- [2] Omicini, A., Zambonelli, F., Klusch, M., Tolksdorf, R., Eds., *Coordination of Internet Agents*, Springer-Verlag, Berlin, 2001.
- [3] The Foundation for Intelligent Physical Agents Web Site, <http://www.fipa.org/>
- [4] Г. Ежич, М. Кушек, И. Люби, Т. Маренич, И. Ловрек, С. Дешич, Б. Деллас, "Mobile-Agent Based Software Management in Grid", *Proceedings of the Workshop on Emerging Technologies for Next Generation GRID*, pp. 345-346, Modena, 2004.
- [5] Gervais, M.-P., Diagne, A., "Enhancing Telecommunication Service Engineering with Mobile Agent Technology and Formal Methods", *IEEE Communications Magazine*, vol. 36, no. 7, pp. 38-43, 1998.
- [6] Knapik, M., Johnson, J., *Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies, and Applications*, McGraw-Hill, New York, 1998.

[7] К. Тржец, Д. Хуленич, "Intelligent Agents for QoS Management", Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002), pp. 1405-1412, Bologna, 2002.

[8] Cockayne, W. R., Zyda, M., Mobile Agents, Prentice Hall, Upper Saddle River, New Jersey, 1997.

[9] D'Inverno, M., Luck, M., Understanding Agent Systems, Springer-Verlag, Berlin, 2001.

[10] Г. Ежич, М. Кушек, С. Дешич, А. Царич, Д. Хуленич, "Multi-Agent System for Remote Software Operation", Lecture Notes in Artificial Intelligence, vol. 2774, pp. 675-682, Springer-Verlag, Berlin, 2003.

[11] М. Кушек, Г. Ежич, И. Люби, К. Млинарич, И. Ловрек, С. Дешич, О. Лабор, А. Царич, Д. Хуленич, "Mobile Agent Based Software Operation and Maintenance", Proceedings of the 7th International Conference on Telecommunications (ConTEL 2003), pp. 601-608, Zagreb, 2003.

[12] Т. Маренич, Г. Ежич, М. Кушек, С. Дешич, "Using Remote Maintenance Shell for Software Testing in the Target Environment", Proceedings of 2nd ICSE Workshop on Remote Analysis and Measurement of Software Systems (RAMSS 2004), pp. 19-24, Edinburgh, 2004.

[13] И. Ловрек, Г. Ежич, М. Кушек, И. Люби, А. Царич, Д. Хуленич, С. Дешич, О. Лабор, "Improving Software Maintenance by Using Agent-based Remote Maintenance Shell", IEEE International Conference on Software Maintenance (ICSM 2003), pp. 440-449, IEEE Computer Society Press, Amsterdam, 2003.

[14] IKV++ Technologies Web Site, <http://www.grasshopper.de/>

[15] The MonALISA Web Site, <http://monalisa.cacr.caltech.edu/>

[16] Foster, I., Keselman, C., Nick, J., Tuecke, S., "Grid Services for Distributed System Integration", IEEE *Computer*, vol. 35, no. 6, pp. 37-46, 2002.

[17] Г. Ежич, М. Кушек, Т. Маренич, И. Люби, И. Ловрек, С. Дешич, К. Тржец, Б. Деллас, "Grid Service Management by Using Remote Maintenance Shell", Lecture Notes in Computer Science, vol. 3270, pp. 136-150, Springer-Verlag, Berlin, 2004.

[18] The Apache ANT Project Web Site, <http://ant.apache.org/>

АДРЕСА АВТОРОВ:

Крунослав Тржец

e-mail: krunoslav.trzec@ericsson.com
Ericsson Nikola Tesla d.d.
Krapinska 45
p.p. 93
HR-10002 Zagreb
Хорватия

Гордан Ежич

e-mail: gordan.jezic@fer.hr
Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu
Unska 3
HR-10000 Zagreb
Хорватия

Марио Кушек

e-mail: mario.kusek@fer.hr
Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu
Unska 3
HR-10000 Zagreb
Хорватия

Саша Дешич

e-mail: sasa.desic@ericsson.com
Ericsson Nikola Tesla d.d.
Krapinska 45
p.p. 93
HR-10002 Zagreb
Хорватия

*Редакция приняла рукопись 21 марта 2005 года.
Перевод: Надежда Племенич*