

© 2012 г. С.С. ВАЛЕЕВ, д-р техн. наук
В.А. МАСЛЕННИКОВ, канд. техн. наук
А.С. КОВТУНЕНКО

Уфимский государственный авиационный технический университет

ПРОЕКТИРОВАНИЕ ПРОМЕЖУТОЧНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ АГЕНТНО-ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЙ ДЛЯ САУ СЛОЖНЫМИ ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ¹

Рассматривается задача разработки программного обеспечения промежуточного уровня для построения распределенных систем управления сложными техническими объектами с применением агентно-ориентированной технологии, что позволяет обеспечить необходимый уровень виртуализации и аппаратной независимости при решении задач управления. Предлагаемый программный комплекс позволяет разрабатывать и отлаживать компоненты систем управления и имитационные модели в едином адресном пространстве.

DEVELOPMENT OF THE MIDDLEWARE FOR COMPLEX TECHNICAL SYSTEM CONTROL / S. S. Valeev, V. A. Maslennikov, A. S. Kovtunenکو (Ufa State Aviation Technical University, 12 Karl Marx st., Ufa, Bashkortostan, 450000, Russia. E-mail: kobtyhehko@ya.ru) The experience in middleware for distributed control systems is considered. Using of agent-oriented development technology allows to provide necessary level of control programs virtualization and hardware-independence. The offered software provide developing and debugging of control system components and simulations in a common address space.

1. Введение

Как известно, для достижения глобальной цели управления в иерархических системах, она декомпозируется на множество взаимосвязанных подцелей, достижение которых обеспечивается на основе взаимодействия гетерогенных вычислительных систем, различающихся как по темпам решения задач, так и по своей сложности.

Так, например, для задач, требующих жесткого реального времени и не требующих больших вычислительных ресурсов аппаратной базой чаще всего служит микроконтроллерный комплекс, алгоритмы же реализуются с привлечением технологий системного программирования.

Рассматривается задача разработки программного комплекса промежуточного уровня для построения распределенных систем управления сложными техническими объектами с применением агентно-ориентированной технологией. В основе предлагаемого подхода лежит идея автономности выполнения задач управления, благодаря чему он позволяет соблюдать устанавливаемые для них требования независимости и изолированности.

¹ Работа выполнена при финансовой поддержке российского фонда поддержки фундаментальных исследований (грант РФФИ № 10-08-00928).

Показана возможность реализации всех задач управления в пределах единой технологии высокого уровня.

2. Постановка задачи

Согласно традиционному подходу, все множество задач управления сложной системой можно разделить на три уровня [1].

- Исполнительный уровень — к нему относятся задачи непосредственного управления аппаратными средствами, съём информации с датчиков. Задачи этого уровня характеризуются повышенным требованиям ко времени исполнения и небольшой вычислительной сложностью.
- Уровень координации — к нему относятся задачи согласования выполнения задач исполнительного уровня, фильтрация данных, а также задачи логического администрирования более низкого уровня. Задачи этого уровня характеризуются большей по сравнению с исполнительным уровнем вычислительной сложностью и более низкими требованиями ко времени выполнения.
- Стратегический уровень — к нему относятся задачи планирования, прогнозирования, моделирования и пр. Стратегические задачи являются наиболее сложными, однако, как правило, не имеют жестких ограничений по времени исполнения.

Очевидно, что проблема управления может на всех уровнях включать в себя как задачи жесткого, так и мягкого реального времени.

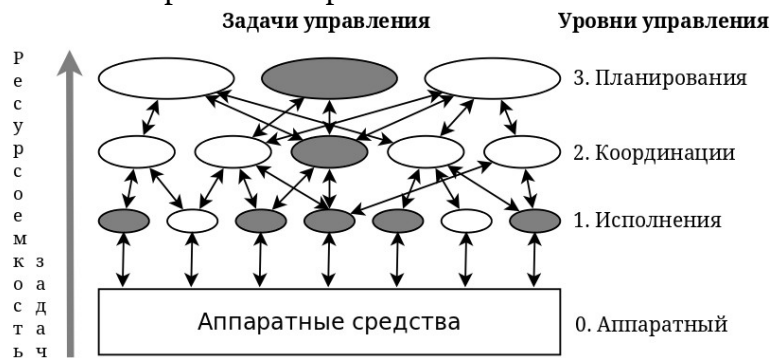


Рисунок 1 — Иерархия задач управления

На рисунке 1 схематически показана иерархия задач управления, отмечено, что ряд задач может иметь ограничения реального времени (выделены серым). С увеличением уровня растет и сложность задач, а следовательно их ресурсоемкость.

Традиционно каждый тип задач реализуется в виде отдельных модулей управляющей системы с привлечением различных технологий разработки. Это существенно осложняет интеграцию отдельных алгоритмов в единую систему управления, поскольку добавляется проблема обеспечения взаимодействия между модулями. Кроме того, усложняется отладка системы, поскольку на этой стадии модули являются уже отлаженными и корректировке почти не подлежат, тогда как проверка их во взаимодействии может выявить ряд системных ошибок.

Поэтому разработка методики и инструментария создания многоуровневых систем управления сложными техническими и организационно-техническими системами все еще является актуальной научно-технической задачей. Можно сформулировать следующие требования для разработки.

- Программные модули управления должны выполняться в едином адресном пространстве. Это позволит вести отладку всей системы параллельно с отладкой

отдельных модулей и избежать системных ошибок на начальных стадиях проектирования и отладки.

- Программные модули управления должны функционировать автономно и изолированно друг от друга. Ошибки других алгоритмов управления не должны влиять на выполнения каждого конкретного алгоритма.
- Аппаратный уровень должен включать в себя все необходимые вычислительные устройства: высокоэффективные узлы, мобильные устройства, микроконтроллеры и пр.
- Технология разработки должна позволять устанавливать для задач требования жесткого и мягкого реального времени.
- Программная реализация алгоритмов управления должна оптимально использовать вычислительные ресурсы сети и гибко адаптироваться к изменениям как ее структуры, так и мощности узлов.

Таким образом, необходимо разработать комплект промежуточного программного обеспечения, который бы позволял создавать системы управления сложными техническими объектами на базе гетерогенной вычислительной сети и удовлетворял вышеперечисленным требованиям.

3. Промежуточное программное обеспечение систем управления сложными техническими системами

Традиционная структура технического обеспечения распределенных вычислительных систем подразумевает наличие аппаратного и трех уровней программного обеспечения (рисунок 2).

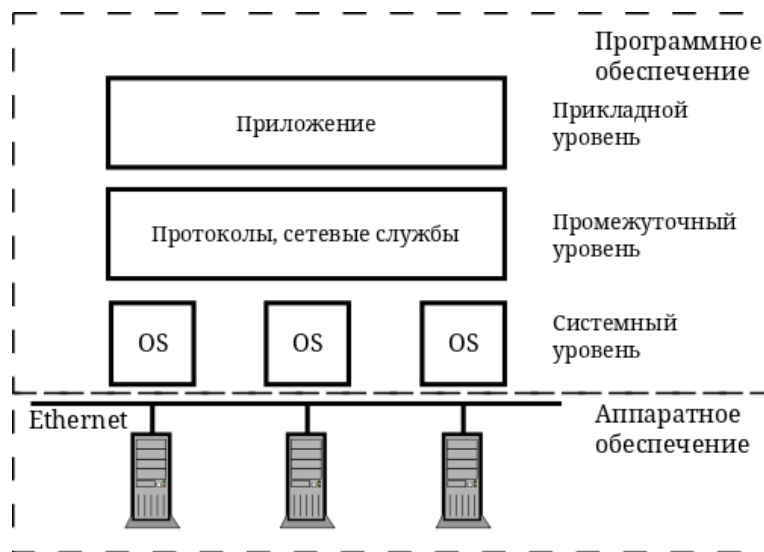


Рисунок 2 — Техническое обеспечение распределенных вычислительных систем

Промежуточное программное обеспечение (ППО) для распределенных систем — это набор библиотек, приложений и сервисов, обеспечивающих необходимый уровень виртуализации в распределенной вычислительной системе для прикладных программ. ППО обеспечивает взаимодействие вычислительных узлов, распределение всех видов нагрузки (CPU, память и сеть), диспетчеризацию заданий, интерфейсы взаимодействия между прикладными программами, а также в том числе оптимизацию работы сложных программно-аппаратных комплексов (рисунок 3) [2].

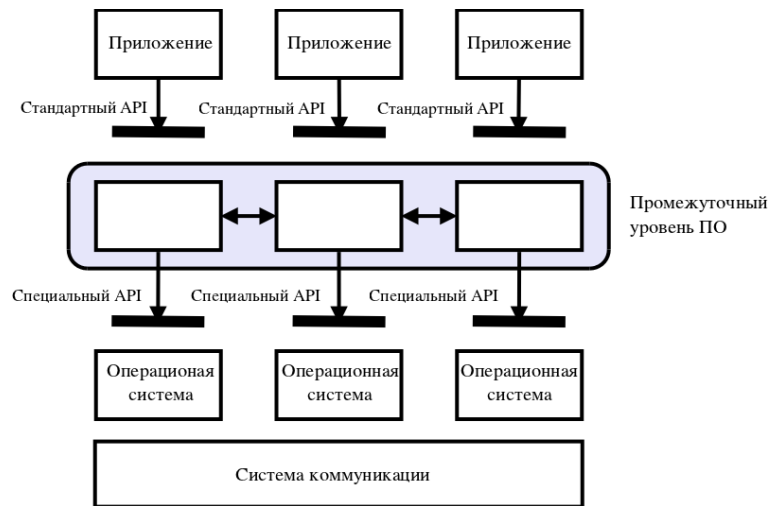


Рисунок 3 — К понятию промежуточного программного обеспечения

По особенностям сетевой инфраструктуры современные системы ППО могут быть классифицированы следующим образом:

- Сети с фиксированной топологией и непредсказуемой загрузкой. Наиболее распространенный случай, к которому в частности относятся корпоративные сети.
- Сети с фиксированной топологией и предсказуемой загрузкой. Этот случай соответствует системам высокой ответственности, таким как системы управления реального времени, взаимодействие в которых должно гарантировать доставку сетевых пакетов в установленное время посредством резервирования ресурсов.
- Сети с нефиксированной топологией и непредсказуемой загрузкой. Этот случай соответствует сетям, которые могут включать в себя мобильные или подвижные устройства, такие как телефоны или планшетные компьютеры. Связь с такими устройствами осуществляется посредством беспроводных технологий, которые допускают непредсказуемые изменения производительности.

По архитектуре приложений и поддерживаемым интерфейсам создания приложений (API) различные типы ППО определяются различными комбинациями следующих свойств.

- Базовые сущности. ППО может быть основано на различных программных сущностях, таких, как объекты, агенты, компоненты и пр., которые отличаются друг друга способом определения, свойствам и моделью взаимодействия.
- Структура. Некоторые сущности, на которых базируется ППО могут иметь предопределенные роли, такие, как клиент или сервер, издатель или абонент, а могут не иметь ролевой дифференциации и быть одного ранга, или же в процессе работы менять свои роли.
- Служебные интерфейсы. Элементарные операции взаимодействия, поддерживаемые ППО могут осуществляться в синхронном и асинхронном режиме.

В функциональной структуре ППО можно выделить следующие основные элементы.

- Протокол передачи задач между узлами.
- Служба диспетчеризации задач.
- Служба оптимизации загрузки узлов.
- Служба доступа к ресурсам памяти.

Кроме вышеперечисленных, ППО может включать и иные компоненты, решающие те или иные задачи комплекса.

Рассматривается реализация ППО на базе агентной платформы Java Agent Development Framework (JADE).

4. Программная платформа JADE

JADE – это программное обеспечение промежуточного слоя (рисунок 3), разработанное компанией TILAB, предназначенное для создания распределенных мультиагентных приложений на основе транспортной архитектуры «точка-точка». Все ресурсы, а также управление могут быть распределены по вычислительной сети. Среда может динамически взаимодействовать с узлами, реализуя концепцию подвижного кода [3]. Java Agent Development Framework (JADE) – программная среда создания многоагентных приложений, поддерживающая FIPA-стандарты для интеллектуальных агентов. Она включает в себя следующие компоненты:

- среду выполнения агентов, в которой агенты регистрируются и под управлением которой функционируют;
- библиотеку классов, используемых для разработки агентной системы;
- набор графических утилит, необходимых для администрирования и наблюдения за жизненным циклом активных агентов.

Коммуникации между узлами не зависят от типа сети (проводная, беспроводная). Они являются полностью симметричными, и каждый узел может, как инициировать запросы, так и отвечать на них. Платформа JADE полностью разработана на языке Java. основополагающими принципами платформы являются:

- Функциональная совместимость – продукт JADE разработан в соответствии со спецификациями FIPA. Как следствие JADE-агенты могут взаимодействовать со сторонними агентами, поддерживающими этот стандарт.
- Портруемость и единообразие – продукт JADE предоставляет гомогенный набор API, которые не зависят ни от базового устройства сети, ни от версии платформы Java.
- Простота использования – набор простых и интуитивно-понятных интерфейсов API прячет сложную логику ПО промежуточного слоя от пользователя.
- Принцип «разрабатывать по средствам» – программистам нет необходимости использовать все возможности, которые предоставляет ПО промежуточного слоя. От программистов не требуется знать что-либо о неиспользуемых функциях платформы. Ни одна из незадействованных функций не создает дополнительные накладные вычислительные расходы.

Платформа JADE включает как программные библиотеки (т.е. наборы Java-классов), требуемые для разработки прикладных агентов, так и среду исполнения, которая предоставляет базовые службы и которая должна быть активна на устройстве до того, как на нем будет исполняться агент.

Каждый экземпляр JADE во время исполнения называется контейнером (так как он «содержит» агентов). Набор всех контейнеров называется платформой и предоставляет однородный слой, который прячет от агентов (а также от разработчиков приложений) сложность и распределенный характер механизмов, расположенных на более низком уровне (аппаратное обеспечение, операционные системы, типы сетей, JVM) (рисунок 4). [4]

Агент – это программа, обладающая независимым поведением и способная обмениваться сообщениями с другими агентами. Агент обладает набором поведений (функций, методов), для обработки заданной реакции на определенные события.

Основной целью данной платформы является обеспечение существования и взаимодействия агентов.

Задачи агентной платформы заключаются в следующем:

- 1) управление жизненными циклами агентов;
- 2) эффективная передача сообщений между агентами.

Архитектура агентной платформы во многом определяется стандартом FIPA.

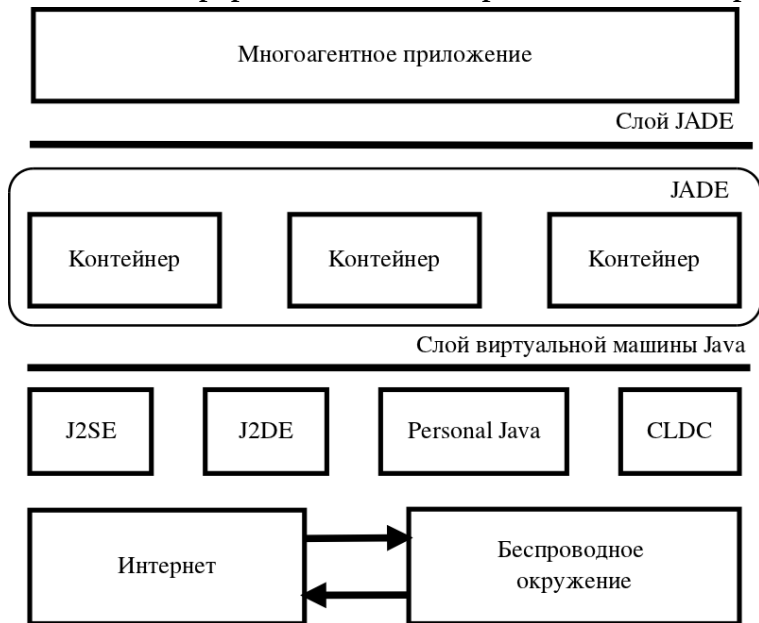


Рисунок 4 - Архитектура JADE

В средах J2SE и Personal Java платформа JADE поддерживает мобильность кода и мобильность состояния. То есть агент может остановить работу на хосте и мигрировать на другой удаленный хост (без необходимости копировать код или устанавливать агента на этом хосте заранее), а затем возобновить исполнение с того места, где он был остановлен. Эта функциональность позволяет организовать распределение вычислительной нагрузки в режиме исполнения без внесения каких-либо изменений в задачу.

Программный комплекс промежуточного уровня для проектирования и построения САУ сложными техническими объектами.

Для реализации задач управления на базе JADE был разработан программный комплекс, архитектура которого показана на рисунке 5.

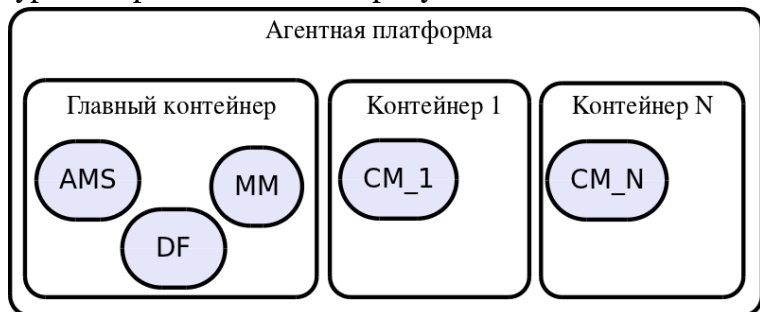


Рисунок 5 - Архитектура разработанного программного комплекса промежуточного уровня.

Основными элементами комплекса являются:

- Агентная платформа, которая обеспечивает существование и взаимодействия агентов. Задачи агентной платформы заключаются в управлении жизненными циклами агентов и в эффективной передаче сообщений между агентами.
- Система управления агентами (AMS) — компонента системы, которая обеспечивает нижний уровень работы с агентами — их создание, уничтожение и перемещение.
- Служба «желтых страниц» (DF) — система обслуживания сервисов платформы. По запросу предоставляет информацию о действующих сервисах и позволяет регистрировать новые.
- Главный менеджер системы (MM) — компонента, обеспечивающая создание и запуск прикладных программ управления, обрабатывающая XML-скрипт с описанием программ и создающая распределенную инфраструктуру управления выполнением прикладных программ.
- Менеджер контейнера (CM) представляет локальный уровень управления выполнением прикладных программ, отвечает за оптимизацию загрузки вычислительного узла.

Программа управления представляется в виде совокупности автономно функционирующих агентов между которыми установлены сигнальные связи. Функционирование агента заключается в преобразовании входящего сигнала с формированием исходящего. Каждому агенту на вход может поступать сигнал от нескольких агентов.

Все агенты имеют общую структуру и схему работы (рисунок 6). Массив входных величин формируется как блочный вектор из поступающих сигналов в той последовательности, в которой их источники шли в адресном списке при создании агента. Функционирование агента заключается в регулярном (с заданным периодом) вызове главной процедуры, также в периодическом (с тем же периодом) опросе агентов-источников о текущем уровне их сигналов. Предполагается, что в главной процедуре рассчитывается новое значение выходного сигнала и предоставляется по запросу агентам, для которых рассматриваемый агент является источником.

С точки зрения разработчика агенты представляют собой java классы, наследующие ряд классов и интерфейсов, которые позволяют агентной платформе создавать их экземпляры в виде агентов. Рассматриваемый комплекс ППО позволяет весь функционал компонента программы управления свести к преобразованию сигнала и поместить в одну периодически вызываемую процедуру. Управление жизненным циклом компонента, передача сигналов между компонентами осуществляется на уровне ППО.

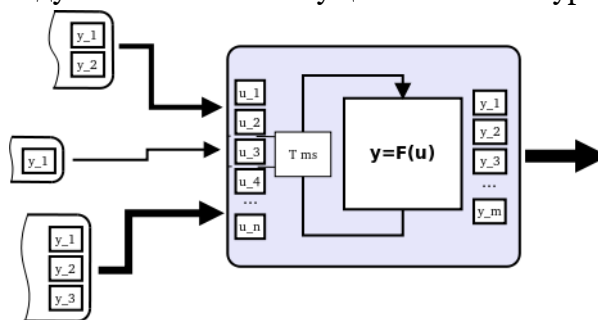


Рисунок 6 — Структура и схема работы агента

Для упрощения процесса проектирования и создания программного обеспечения систем управления в рассматриваемом программном комплексе был разработан специальный инструментарий. Он представляет собой совокупность следующих элементов:

- Набор готовых настраиваемых компонентов,

- Встроенный XML язык создания прикладных программ, позволяющий в декларативной форме описывать структуру системы управления.
- Редактор файлов структурного описания программ управления.

Благодаря этому создание программного обеспечения системы управления заключается в построении структурной схемы управляющей программы в виде совокупности «виртуальных устройств» и сигнальных связей между ними. Сигнальное представление управляющей программы почти точно воспроизводит обычную структурную схему системы управления. Источником сигналов являются как правило регистрирующие компоненты, которые осуществляют взаимодействие с датчиками или с элементами управления (например кнопками). Вывод сигналов осуществляется на компоненты, взаимодействующие непосредственно с устройствами-актуаторами, передающие реальному процессу управляющие воздействия (например сервоприводами).

Программа может быть как составлена из готовых компонентов, так и дополнена новыми, если в силу специфики задачи управления сформировать ее из предлагаемого набора компонент не представляется возможным.

На данном этапе разработки система снабжена набором компонент, соответствующим большинству типовых звеньев.

- Линейное звено произвольного порядка. С помощью параметров может быть сконфигурировано как:
 - динамическое звено с возможностью задания как в форме передаточной функции (коэффициенты числителя и знаменателя, допускает только скалярный сигнал), так и в уравнениях состояния (матрицы A, B, C, D) с установкой начального состояния;

$$(1) \quad \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^k, \quad y \in \mathbb{R}^m$$

где n, k, m — размерность звена, количество входов и количество выходов соответственно.

- скалярный коэффициент;

$$(2) \quad y = d u$$

- свертка векторного сигнала в скалярный (скалярное произведение двух векторов);

$$(3) \quad y = d^T u$$

- линейный оператор, заданный матрицей общего вида (умножение вектора на матрицу);

$$(4) \quad y = Du$$

- Набор модельных источников сигналов:

- ступенчатое воздействие;
- константа;
- гармонический сигнал;
- источник белого и розового шума;
- Набор компонент графического интерфейса пользователя:
 - отображение временной диаграммы сигналов;
 - «виртуальная кнопка» - управляемый источник ступенчатого сигнала;
 - управляемый источник «непрерывного сигнала», снабженный элементом типа «слайдер»

Использование разработанного ППО позволяет создавать и эксплуатировать программы управления совместно со всякого рода имитационными моделями, в том числе и ресурсоемкими, поскольку в качестве вычислительного узла может быть привлечена высокопроизводительная вычислительная система. [5] Автономность выполнения агентов позволяют говорить об отказоустойчивости программы управления, поскольку ошибка одного или нескольких агентов не влечет за собой остановку всей системы, а лишь снижает её эффективность.

Для соблюдения требований реального времени предлагается модификация стандартного загрузчика агентной платформы JADE. В соответствие со спецификацией RTSJ [6] язык java поддерживает все стандарты реального времени и широко используется во встраиваемых системах. Таким образом, при необходимости в системе могут быть созданы компоненты, удовлетворяющие требованиям жесткого реального времени.

Очевидно, что большая часть компонентов распределенной системы управления будет жестко привязана к устройству. Это относится прежде всего к тем из них, что работают с драйверами аппаратного обеспечения (контроллеры, датчики), составляют графический интерфейс пользователя (временные диаграммы, экранные элементы управления: кнопки, слайдеры), а также обеспечивают интерфейс с БД. В случаях, когда в состав программы управления входят вычислительные алгоритмы (принятия решения, прогнозирования, имитационного моделирования, статистической обработки данных), то во-первых можно говорить о вариативной составляющей нагрузки, а во-вторых — возникает дополнительная задача оптимизации распределения ресурсов системы за счет перераспределения нагрузки между узлами.

Механизм перераспределения нагрузки основан на понятии эффективности вычислений. Все агенты, не имеющие привязки к узлу на каждой итерации рабочего цикла рассчитывают эффективность своей работы как время, потраченное на целевые расчеты по отношению к периоду. Если эффективность попадает в установленные критичные пределы, агент инициирует механизм поиска подходящего узла для перемещения. В этом механизме задействованы менеджеры контейнеров (СМ), а также главный менеджер (ММ). Если удастся найти узел, на котором у данного агента эффективность не будет критично низкой, агенту дается разрешение на перемещение и он мигрирует в другой контейнер. Таким образом в условиях гетерогенной вычислительной системы и динамически изменяющихся доступных ресурсах (производительности узлов) вычислительная нагрузка будет распределена оптимальным образом.

5. Выводы

Рассмотрена задача построения промежуточного программного обеспечения распределенных систем управления сложными техническими объектами, сформулированы требования к ППО такого рода и поставлена задача разработки. Предложена реализация

ППО для распределенных систем управления. Показано, что использование агентных технологий в качестве базовых для реализации программ управления позволяет создавать системы гибкими, отказоустойчивыми.

Показана возможность и предложен инструментарий для упрощения процесса создания программ управления за счет использования типовых решений. Разработанное ППО позволяет использовать в системах управления имитационное моделирование и вовлекать в единое адресное пространство компонентов системы управления как мобильные устройства, так и высокопроизводительные узлы, а также оптимально распределять нагрузку на узлы.

Кроме того, использование агентных технологий в управлении дает возможность совершенствования алгоритмов, вовлечения в процесс управления концепций искусственного интеллекта, обучения, адаптации и самоорганизации.

СПИСОК ЛИТЕРАТУРЫ

1. *М. Месарович, Д. Мако, И. Такахара* теория иерархических многоуровневых систем. М.: Издательство «Мир», 1973.
2. *Gordon S Blair, Geoff Coulson, Phillippe Robin, Michael Papathomas*. An architecture for next generation middleware // Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing. Springer-Verlag, 1998. P. 191-206
3. *A. Fuggetta, G. P. Picco, and G. Vigna*. Understanding Code Mobility // IEEE Trans. on Software Engineering, 24(5), 1998.
4. *Fabio L. Bellifemine, Giovanni Caire, Dominic Greenwood*. Developing Multi-Agent Systems with JADE: John Wiley&Sons, Ltd, 2007
5. *С. С. Валеев, А. С. Ковтуненко, Р. М. Сайфуллин*. Создание распределенных систем управления на базе многоагентных технологий реального времени // Актуальные проблемы науки и техники. Том 1 - Информационные и инфокоммуникационные технологии, естественные науки: Сборник трудов седьмой Всероссийской зимней школы-семинара аспирантов и молодых ученых. Уфа – УГАТУ, 2012. С. 211-214
6. RTSJ 1.0.2 specification. http://www.rtsj.org/specjavadoc/book_index.html