

УДК 004.89

РАЗРАБОТКА МНОГОАГЕНТНЫХ СИСТЕМ РАСПРЕДЕЛЕННОГО РЕШЕНИЯ ЭНЕРГЕТИЧЕСКИХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ АГЕНТНЫХ СЦЕНАРИЕВ

Массель Людмила Васильевна,

д-р техн. наук, зав. лабораторией информационных технологий в энергетике
Института систем энергетики им. Л.А. Мелентьева СО РАН, Россия, 664033,
г. Иркутск, ул. Лермонтова, 130. E-mail: massel@isem.sei.irk.ru

Гальперов Василий Ильич,

аспирант лаборатории информационных технологий в энергетике
Института систем энергетики им. Л.А. Мелентьева СО РАН,
Россия, 664033, г. Иркутск, ул. Лермонтова, 130. E-mail: galperov@gmail.com

Актуальность работы определяется, с одной стороны, развитием концепции «интеллектуальной сети» (Smart Grid), одним из важных аспектов которой является использование многоагентных технологий, с другой – малым количеством практических подходов к реализации многоагентных систем. В современных условиях, для функционирования и управления электроэнергетических систем требуется создание расчетной модели для схем большой размерности на базе методов оценивания состояния. Такие схемы не полностью наблюдаемы, возможно искажение данных, плохая их синхронизация и, как следствие, принятие неправильных решений, формируемых на базе расчетной модели. Существует необходимость в разработке новых методов и программных средств для оценивания состояний, которые позволят исключить эти проблемы.

Цель работы: провести обзор агентных решений, имеющихся в настоящее время на рынке информационных технологий; рассмотреть недостатки существующих систем и предложить основанный на использовании агентных сценариев подход к разработке многоагентных систем для решения энергетических задач, который позволит устранить эти недостатки.

Методы исследования. В настоящее время агентный подход не применяется для разработки систем, направленных на решение задач в области электроэнергетики и оценивания состояний электроэнергетических систем в частности. Предлагаемый подход к разработке многоагентных систем должен обеспечивать возможность взаимодействия агентов, созданных на разных программных платформах, создания новых и редактирование существующих алгоритмов в системе посредством агентных сценариев.

Результаты. Разработан подход к созданию многоагентных систем для оценивания состояний электроэнергетических систем с возможностью межплатформенных взаимодействий и агентных сценариев. Предлагаемый подход был апробирован на тестовой системе, и в данный момент разрабатывается система для оценивания состояний электроэнергетических систем.

Ключевые слова:

Агент, многоагентные системы, оценивание состояний ЭЭС, агентные сценарии, Smart Grid.

Введение

Развитие рыночных отношений в электроэнергетике России привело к появлению новых задач, для решения которых необходима расчетная модель текущего режима электроэнергетической системы, получаемая на основе данных телеизмерений с помощью методов оценивания состояния (ОС) [1, 2].

Одной из тенденций развития современных энергетических систем является концепция интеллектуальных энергетических систем (Smart Grid), которая нацелена на создание электрических сетей, удовлетворяющих будущим требованиям по энергоэффективному и экономичному функционированию энергосистемы за счет скоординированного управления и при помощи современных двусторонних коммуникаций между элементами электрических сетей, электрическими станциями, аккумулирующими источниками и потребителями [3].

Существенно улучшить свойства решения задачи оценивания состояния позволяет использование измерений, поступающих от устройств измерения комплексных электрических величин – PMU (Phasor Measurement Units) [4]. Измерения, поступаю-

щие от PMU, более полно отражают режим рабочей схемы электроэнергетических систем (ЭЭС). Кроме этого, важное место в этой концепции занимает *многоагентный подход* к созданию автоматизированных программных средств управления электроэнергетическими системами. Одними из основных направлений работ по развитию автоматизированной системы управления режимами являются разработка алгоритмов выявления предаварийных состояний энергосистем и диагностики электротехнического оборудования на основе методов оценивания состояний и параметрической идентификации и создание систем распределенного расчета режимов энергосистем. Применение многоагентного подхода позволит разработать оперативную и гибкую в настройке систему оценивания состояний ЭЭС [5].

К сожалению, в отношении агентных технологий практически нет четких методов разработки и алгоритмов функционирования применительно к задачам энергетики. Основные достижения в этой части в основном ориентируются на аспекты теоретической реализации и пока далеки от практики. В статье рассматриваются существующие решения, их недостатки и предлагается подход, направленный на их устранение.

Определение агента

Под агентом принято понимать автономный процесс, способный реагировать на среду исполнения и вызывать в ней изменения, возможно, в кооперации с пользователями или другими агентами. Нередко агенты понимаются как вычислительные единицы, поддерживающие локальные состояния и параллельные вычисления, а также способные в процессах коммуникации достигать состояния других агентов и автоматически выполнять действия в некоторых условиях среды [6, 7]. Также под агентом понимается «любая сущность, которая находится в некоторой среде, воспринимает ее посредством сенсоров, получая данные, которые отражают события, происходящие в среде, интерпретирует эти данные и действует на среду посредством эффекторов». Таким образом, здесь выделяются четыре исходных фактора, образующих агента: среда, восприятие, интерпретация, действие [8].

Как правило, агенты действуют параллельно. Контроль и обработка данных логически разделены между различными элементами. Чтобы решать проблемы совместно, агенты должны общаться, координироваться и вести переговоры друг с другом, как только они оказываются в конфликтной ситуации. Системы, содержащие группу агентов, которые могут взаимодействовать между собой, и называются мультиагентными системами. Основные преимущества использования многоагентных систем, это: распределенные вычисления, масштабирование и автономность. Агенты должны быть распределены по локальной сети для обеспечения децентрализованной обработки данных. Они должны уметь общаться между собой, обмениваться информацией и кооперироваться для выполнения задач.

Существующие подходы к разработке многоагентных систем

Для организации процесса распределения задачи в многоагентных системах создается либо система распределенного решения проблемы, либо децентрализованная система искусственного интеллекта. В случае использования последней распределение заданий происходит в процессе взаимодействия агентов и носит больше спонтанный характер [9].

Децентрализованная система искусственного интеллекта используется в основном для агентного моделирования в том случае, когда больше важен не финальный результат, а организация процесса. Например, моделирование поведения толпы при эвакуации или же торги между продавцом и покупателем на рынке. Тогда каждый агент будет преследовать свои собственные цели, которые могут быть полностью противоположны интересам других агентов.

Одним из самых известных программных продуктов, реализующих данную концепцию, является AnyLogic. Это инструмент имитационного моде-

лирования, который позволяет проектировать агентные, системно-динамические, дискретно-событийные и «многоподходные» модели [10]. Также AnyLogic предоставляет широкий спектр отчетов и статистику по работе модели. Для создания моделей в основном используется графический редактор, где пользователь визуально описывает все составные элементы будущей модели. Для задания логики поведения отдельных элементов созданной модели применяется объектно-ориентированный язык программирования Java.

Однако же программные средства, придерживающиеся данной концепции, подходят исключительно для моделирования процессов и не могут помочь при решении каких-либо вычислительных задач. Если нам необходимо решать комплексные вычислительные задачи, то в такой ситуации агенты должны действовать сообща, чтобы выполнить задачу максимально качественно и в наиболее короткие сроки. Также во время работы системы в нее могут входить новые агенты, а другие перестают работать. Агенту необходимо получать информацию о других агентах, находящихся в системе, чтобы знать, с кем можно взаимодействовать. В таком случае выполняется процесс декомпозиции глобальной задачи и обратный процесс агрегирования найденных решений происходит под управлением некоторого единого «центра». При этом многоагентная система проектируется строго сверху вниз, исходя из ролей, определенных для агентов, и результатов разбиения глобальной задачи на подзадачи. Несмотря на большое количество программных реализаций для этого типа задач многие из них уже не развиваются и не поддерживаются разработчиками.

Кроме этого, одной из современных технологий для построения многоагентных систем является SOA (Service Oriented Architecture). Преимущества использования Web-сервисов для реализации агентов рассмотрены в [11–15]. Однако данный подход применялся лишь в единичных случаях.

Для разработки стандартов в области создания многоагентных систем была сформирована организация FIPA (Foundation for Intelligent Physical) [16]. Было достаточно много разработок, которые поддерживали стандарты, предложенные FIPA, среди них такие, как: Java Intelligent Agent Componentware, The SPADE Multiagent and Organizations Platform, JACK Intelligent Agents, The Fipa-OS agent platform, AgentService, Zeus Agent Building Toolkit и другие.

Одной из широко распространенных программных сред для разработки многоагентных систем является JADE, написанная на языке Java [17]. JADE использует концепцию распределенного решения задач. Основой данной системы является программная среда, без нее невозможно существование агентов. Внутри среды формируются контейнеры, в которые затем помещаются агенты. После запуска каждый агент должен передать данные о себе в один из контейнеров, чтобы зарегистри-

стрироваться в системе. Далее программная среда будет наблюдать за работой всей системы и при необходимости производить требуемые управляющие воздействия на отдельных агентов. Между собой агенты обмениваются при помощи сообщений на языке ACL, который поддерживает стандарт FIPA [18]. Сообщения могут носить разный характер: запрос, информирование, передача данных и т. д. Во время своей работы агент накапливает сообщения, которые приходят ему от других агентов и по очереди занимается их обработкой.

JADE на данный момент является одним из наиболее активно развивающихся «Фреймворков» для разработки многоагентных систем. Однако привязка исключительно к платформе Java ограничивает ее использование. В случае, если исходная программа создана на другом языке программирования, могут возникнуть проблемы при организации взаимодействия между JADE-элементами и компонентами самой программы. Также есть проблема модернизации существующих систем, уже использующих платформу JADE. Для изменения алгоритма работы системы необходимо вмешательство специалиста-программиста. Это мешает пользователю вносить корректировки в работу какого-либо конкретного алгоритма, заложенного в систему. Таким образом, на данный момент существует очень небольшое количество платформ, позволяющих организовывать работу многоагентных систем. Большинство созданных ранее систем уже не поддерживаются и не развиваются, за исключением единичных представителей.

Именно поэтому разработка новых подходов к реализации многоагентных систем является актуальной и востребованной задачей.

Алгоритм поведения агентов

Конкретный агент не изолирован от других агентов. Предлагается осуществлять обмен информацией между отдельными агентами системы посредством передачи XML-файлов по сети. Для передачи данных по локальной сети очень распространен стек протоколов TCP/IP. Большинство программных платформ могут использовать TCP/IP для передачи данных по сети. Таким образом, при помощи TCP/IP мы можем обмениваться информацией между программами, написанными на разных программных платформах. То же самое касается и агентов. В том случае, если какую-то задачу проще реализовать и рассчитывать на конкретной платформе, можно использовать агента, работающего именно под ней, – он точно так же сможет взаимодействовать с другими агентами. XML-файлы, которые используются в системе, состоят из двух частей. В первой части находится информация о местонахождении агента в локальной сети, чтобы агент-получатель знал, по какому адресу ему необходимо отправлять ответ [19].

Для этого вся информация, касающаяся существующих в системе агентов, хранится централизованно в главном модуле. Он составляет и редактирует список активных агентов, следит за ходом выполнения работы, а также хранит исходные данные для расчетов, промежуточные и итоговые

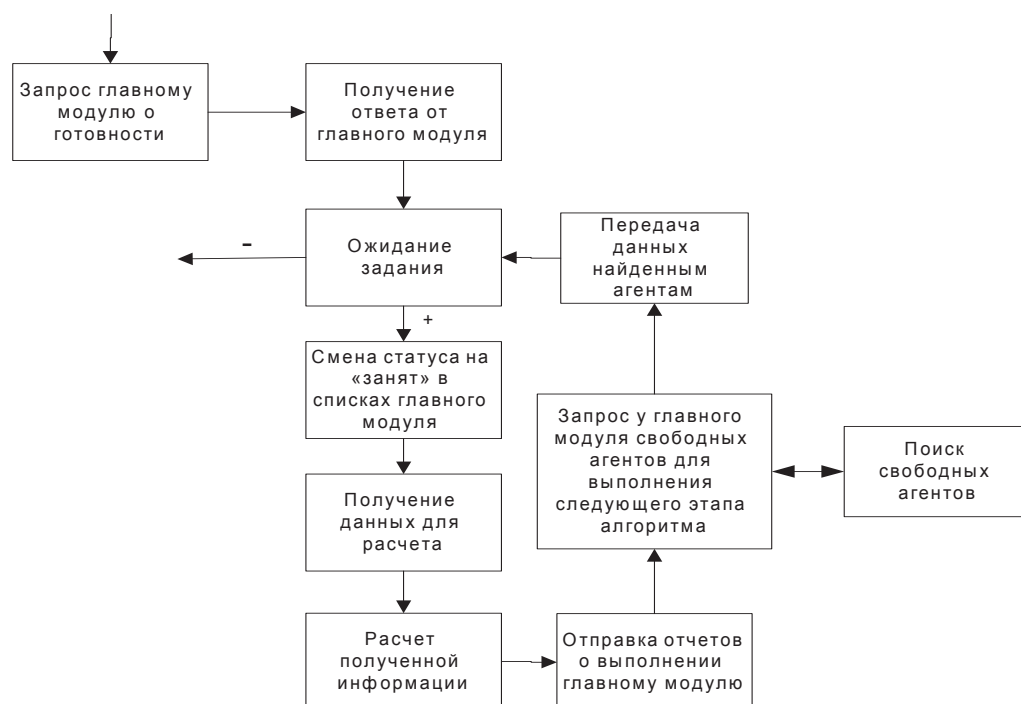


Рис. 1. Схема общего алгоритма работы агентов

Fig. 1. Diagram of general algorithm of agent functioning

результаты работы агентов. Про изменение в своем состоянии агент должен известить главный модуль, чтобы тот обладал актуальной информацией о состоянии системы. В том случае, если агент выполняет работу дольше, чем планировалось, главный модуль отправляет запрос, чтобы агент передал отчет о ходе выполнения работы. В случае, если ответ от агента не приходит (агент «завис», отключился и т. д.), главный модуль удаляет его из списка агентов и перепоручает его работу другому свободному в данный момент агенту или ждет пока таковой освободится.

На рис. 1 представлен общий алгоритм работы агента. После того как агент включился и готов к работе, он направляет информацию о себе главному модулю, чтобы тот внес его в список активных агентов и в дальнейшем к нему могли обращаться другие агенты. Далее агент ждет поступления новой задачи и после этого начинает выполнять вычисления. После окончания вычислений агент формирует об этом главный модуль. Когда агент выполнит свою часть алгоритма, он должен передать данные другому агенту, который продолжит вычисления. До начала передачи этого агента необходимо найти. Его поисками будет заниматься тот агент, который только что закончил свою часть алгоритма. Он запрашивает у главного модуля список свободных на данный момент агентов и отправляет каждому запрос. Как только агентом получен ответ, информация передается главному модулю, а данные для расчета отсылаются найденному агенту. Выход из алгоритма осуществляется после завершения вычислений.

Распределенное решение задач

Одно из основных преимуществ многоагентных систем – распределенная обработка данных. Нецелесообразным является использование многоагентных систем для реализации линейных алго-

ритмов решения задач, поскольку затраты времени на пересылку данных по сети будут лишь тормозить процесс расчета задач. Однако в случае, когда алгоритм позволяет проводить какие-либо вычисления параллельно, распределенная обработка может значительно сократить время расчетов. Поэтому при поиске агентов могут возникнуть сложные ситуации, требующие кооперации между агентами. Основные возможные ситуации, которые могут возникнуть при поиске агентов, указаны на рис. 2.

В первом случае (а) агенту необходимо найти другого агента. В данной ситуации никаких сложностей нет. Первый же ответивший на запрос агент становится искомым и приступает к выполнению задания.

Вторая ситуация (б) ненамного отличается от первой. Здесь один агент должен найти двух или более агентов, чтобы те продолжили решать задачу, выполняя параллельные вычисления. Важно обработать ситуацию, когда из требуемого количества агентов свободны только несколько. В таком случае следует дать задание свободным агентам, после этого запросить у главного модуля обновленный список и продолжить поиски до тех пор, пока не найдется свободный.

Самая проблематичная ситуация, когда нескольким агентам необходимо найти одного (в). Это может произойти, если распределенный этап алгоритма сменяется линейным этапом. В этом случае агентам необходимо взаимодействовать друг с другом, чтобы определить следующего агента. Каждый должен знать координаты остальных агентов, чтобы информировать друг друга о результатах поиска. Все агенты начинают поиск в тот момент, когда закончат выполнять свою задачу. Тот агент, который первым нашел свободного агента, извещает остальных об этом и пересылает им адрес найденного агента, чтобы они знали, куда переда-

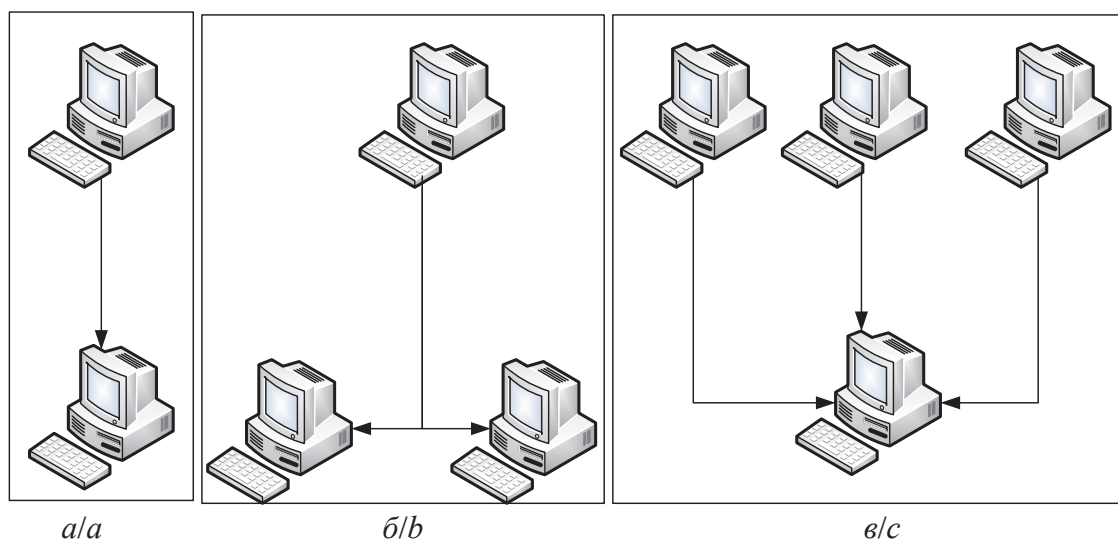


Рис. 2. Ситуации, которые могут возникнуть при поиске агентов

Fig. 2. Situations occurring at agent searching

вать данные. Может произойти так, что два агента одновременно обнаружили свободного. Поэтому перед тем как передавать данные найденному агенту, агенты-поисковики должны достигнуть единого мнения в том, какого из найденных агентов они будут использовать. Определяется, какой из агентов был найден раньше, тому и отдается предпочтение. Когда агенты-поисковики окончательно утвердили кандидатуру следующего агента, они передают ему данные для расчетов.

Агентные сценарии

Агентные сценарии предлагаются для редактирования алгоритма решения задачи без участия программистов. Сценарии формируются пользователями-экспертами. Каждый агент в системе выполняет определенный этап того или иного алгоритма. Однако возможны случаи, когда необходимо внести изменения в процесс решения задач.

Для того чтобы не приходилось вносить изменения в главный модуль, и чтобы пользователь мог сам редактировать существующий алгоритм, предлагается использовать сценарии для агентов. После включения агент должен передать главному модулю не только информацию о своем местонахождении в сети, но и о тех задачах, которые он умеет решать.

Таким образом, составляется список различных типов агентов. Когда пользователю нужно провести расчеты, он может собственноручно выстраивать порядок, в котором будут вызваны те или иные агенты. При необходимости он сможет отредактировать существующий алгоритм, добавив новые этапы расчетов или убрать существующие, чтобы получить промежуточные результаты. Пример агентного сценария для решения задачи оценивания состояния ЭЭС приведен на рис. 3. Там изображены два сценария оценивания состояния ЭЭС с разным методом декомпозиции исходной схемы.

В некоторых ситуациях агенту для решения задачи необходимо получить от пользователя дополнительные параметры. В таком случае агенты могут передавать не только информацию о местонахождении и решаемых задачах, но и о параметрах, которые требуются от пользователя. На основании полученных от агента данных главный модуль сгенерирует форму для ввода значений и передаст ее пользователю. Форма должна состоять из графических элементов интерфейса, таких как: списки, поля ввода текста, поля ввода чисел и т. д. После ввода необходимой информации пользователь отправляет ее главному модулю. Главный модуль проверяет корректность введенной пользователем информации на основании тех ограничений, которые ему передал агент. После проверки данные возвращаются агенту, и на их основании он выполняет требуемые вычисления.

Основное преимущество применения многоагентного подхода – это возможность организовать

распределенную обработку данных. При создании нового или редактировании существующего сценария у пользователя должна быть возможность указать те места, в которых данные можно разделить между несколькими агентами одного типа. Также в сценарии можно будет задать максимальное и минимальное количество агентов, между которыми будет распределена задача. Данная ситуация иллюстрируется на рис. 4.

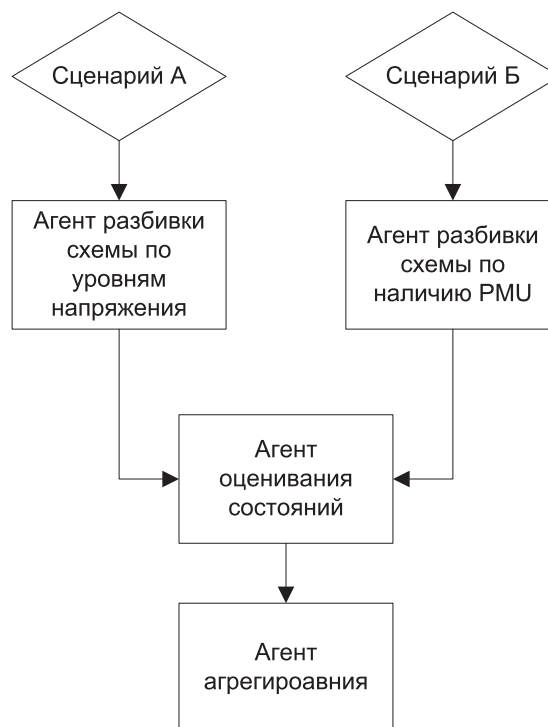


Рис. 3. Агентные сценарии для задачи оценивания состояния ЭЭС

Fig. 3. Agent scenarios for estimating electric power system

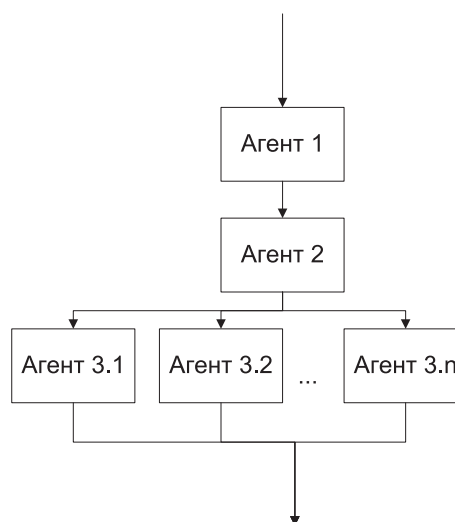


Рис. 4. Распределение задачи между агентами одного типа

Fig. 4. Task distribution among one-type agents

Для создания более «гибких» алгоритмов также предлагается добавлять в сценарии условные операторы и циклы (рис. 5). В таком случае полученные на текущем этапе результаты будут проходить логическую проверку, по итогам которой будет выполняться то или иное действие.

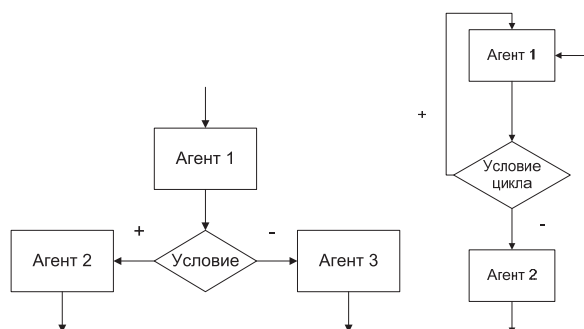


Рис. 5. Условные операторы и циклы в агентных сценариях

Fig. 5. Conditional constructs and cycles in agent scenarios

Применение предложенного подхода для решения распределенной задачи оценивания состояний электроэнергетических систем

На рис. 6 приведена общая схема многоагентной системы для оценивания состояния ЭЭС [20].

Система должна обеспечивать возможность выполнения следующих функций:

- деление расчетной схемы ЭЭС на подсистемы;
- оценивание состояния подсистем;
- решение координационных задач между подсистемами;
- агрегирование результатов расчетов.
- обеспечение возможности работы агентов в локальной сети.

Сначала клиент передает главному модулю системы исходную схему ЭЭС для расчетов, после этого начинается поиск агента.

На первом этапе производится декомпозиция расчетной схемы по уровням узловых напряжений. Границами областей являются узлы, смежные с узлами класса напряжения данной области.

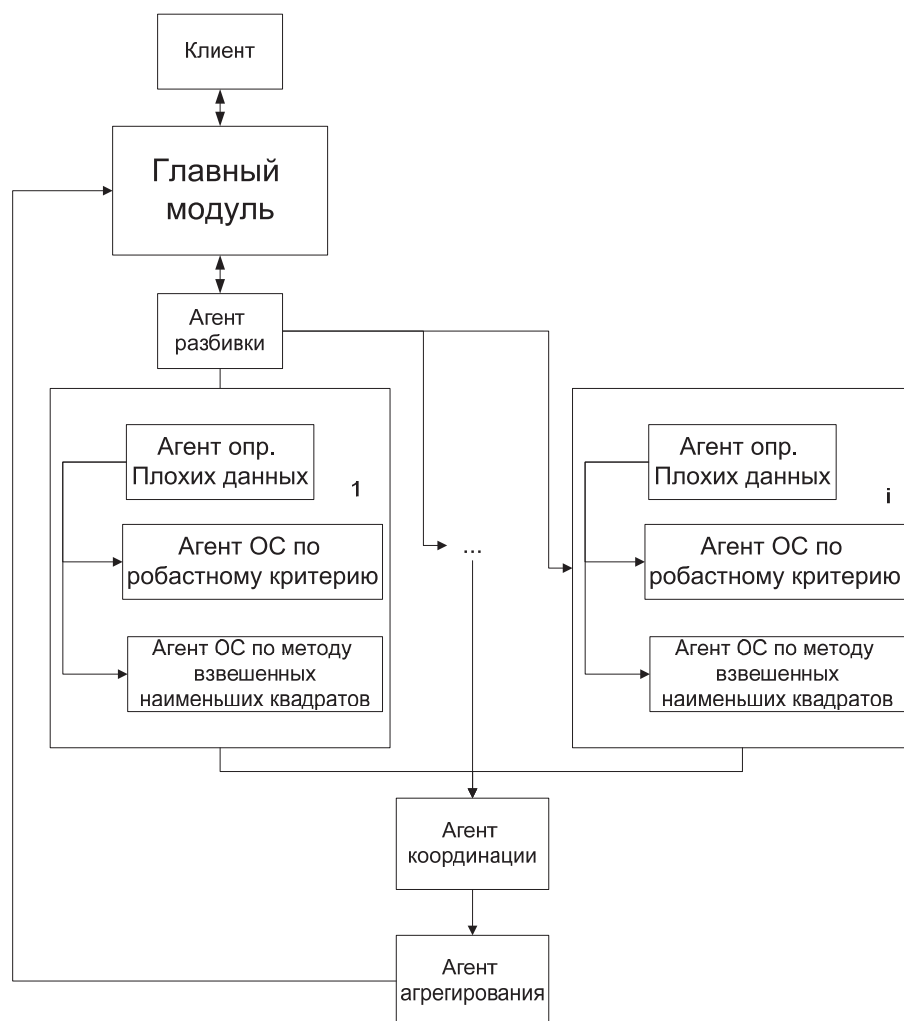


Рис. 6. Схема взаимодействия агентов в многоагентной системе оценивания состояния ЭЭС

Fig. 6. Diagram of agent relation in multi-agent system for estimating the state of electric power system

Так, для области класса напряжения 750–500 кВ граничными являются узлы напряжением 220 кВ, и наоборот.

Далее каждая из полученных схем распределяется между агентами оценивания состояния и рассчитывается параллельно. Расчет начинается с области самого высокого уровня напряжения (750–500 кВ) для каждой подсистемы первого уровня декомпозиции. Как правило, эта часть схемы хорошо обеспечена телеизмерениями высокой точности и содержит базисный узел. Алгоритм оценивания состояния по подсистемам с граничными узлами состоит в следующем:

- Для каждой области, включающей граничные узлы, решается задача обнаружения плохих данных методом контрольных уравнений.
- В случае обнаружения всех плохих данных или их отсутствия решается задача оценивания состояния методом взвешенных наименьших квадратов.
- В случае невозможности определения всех ошибочных телеизмерений, и соответственно невозможности обнаружения плохих данных, выполняется оценивание состояния с помощью робастного критерия (подавления плохих данных).
- Последовательно рассчитываются остальные фрагменты схемы, ранжированные по уровням напряжений (220, 110 кВ и т. д.), всякий раз в качестве базисного узла выбирается узел, граничный с областью более высокого уровня напряжений. Оценки граничных переменных

вектора состояния, полученные на верхнем уровне декомпозиции, фиксируются.

На следующем этапе все схемы необходимо вновь соединить в одну, однако для начала необходимо провести согласования данных в граничных узлах, этим занимается агент координации. Если согласование не было достигнуто, то в схему вносятся корректировки и вновь производится оценивание состояния. В случае если согласование достигнуто, схемы передаются агенту агрегации, который вновь объединяет их в исходную схему и возвращает клиенту.

Заключение

Выполнен обзор существующих на данный момент агентных решений, выделены их области применения и указаны их недостатки. Предложен подход к разработке многоагентных систем, позволяющий избавиться от этих недостатков, который включает в себя: межплатформенное взаимодействие агентов в локальной сети через протоколы TCP/IP, обмен данными посредством XML-сообщений, а также возможность добавлять в систему новых агентов и редактировать существующие алгоритмы при помощи агентных сценариев, задаваемых пользователями-энергетиками. В настоящее время предложенный подход тестируется на практике при создании многоагентной системы оценивания состояний ЭЭС [19].

Работа выполнена при частичной финансовой поддержке грантов РФФИ № 15-07-01284, 13-07-00140, 14-07-00116, № 15-57-04074 Бел_мол_а а также грантов Программы Президиума РАН № 229.

СПИСОК ЛИТЕРАТУРЫ

1. Оценивание состояния в электроэнергетике / А.З. Гамм, Л.Н. Герасимов, И.И. Голуб, Ю.А. Гришин, И.Н. Колосок. – М.: Наука, 1983. – 302 с.
2. Гамм А.З. Статистические методы оценивания состояния электроэнергетических систем. – М.: Наука, 1976. – 220 с.
3. Кобец Б.Б., Волкова И.О. Инновационное развитие электроэнергетики на базе концепции Smart Grid. – М.: ИАЦ Энергия, 2010. – 208 с.
4. Phadke G. Synchronized Phasor Measurements. A Historical Overview. // IEEE/PES Transmission and Distribution Conference. – 2002. – № 1. – С. 476–479.
5. Постановка задачи разработки мультиагентной системы для оценивания состояний ЭЭС с учетом структурной и функциональной декомпозиции / В.И. Гальперов, И.Н. Колосок, Л.В. Массель, А.С. Пальцев // Информационные и математические технологии в науке и управлении: Труды XVIII Байкальской Всеросс. конф. Ч. III. – Иркутск: ИСЭМ СО РАН, 2013. – С. 231–234.
6. Городецкий В.И. Многоагентные системы: современное состояние исследований и перспективы применения // Новости искусственного интеллекта. – 1996. – № 1. – С. 44–59.
7. Городецкий В.И. Многоагентные системы: основные свойства и модели координации поведения // Информационные технологии и вычислительные системы. – 1998. – № 1. – С. 22–34.
8. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд. / пер. с англ. – М.: ИД «Вильямс», 2006. – 1408 с.
9. Многоагентные системы. 2009–2014. URL: <http://www.aiportal.ru/articles/multiagent-systems/multiagent-systems.html> (дата обращения: 02.09.2014).
10. AnyLogic. Многоподходное имитационное моделирование // AnyLogic. URL: <http://www.anylogic.ru/features> (дата обращения: 02.09.2014).
11. Odell J., Giorgini P., Muller J.P. Agent-Oriented Software Engineering // 4th International Workshop, AOSE 2003. – New York, USA, July 2003. – P. 47–59.
12. Agent-Oriented Information Systems II / P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low // 6th International Bi-Conference Workshop, AOIS 2004. Riga, Latvia, June 8, 2004; New York, USA, July 20, 2004. – P. 110–124.
13. Extending Web Services Technologies: the Use of Multi-Agent Approaches / L. Cavedon, Z. Maamar, D. Martin, B. Benatallah // Multiagent Systems, Artificial Societies, and Simulated Organizations. – New York: Springer US, 2004. – P. 29–52.
14. Фаргышев Д.А., Черноусова Е.С., Черноусов А.В. Подход к разработке многоагентной распределенной интеллектуальной информационной системы для исследований в энергетике // Вычислительные технологии. – 2008. – Т. 13. Спец. выпуск 1. – С. 108–115.
15. Фаргышев Д.А., Черноусов А.В. Методы использования Web-сервисов для построения вычислительной ИТ-инфраструктуры системных исследований в энергетике // Информационные и математические технологии в науке и управлении: Тр. XII Байк. Всеросс. конф. Ч. 2. – Иркутск: ИСЭМ СО РАН, 2007. – С. 46–54.

16. FIPA Abstract Architecture Specification. URL: <http://www.fipa.org/specs/fipa00001/SC00001L.html> (дата обращения: 07.09.2014).
17. JAVA Agent Development Framework / Jade. 2015. URL: <http://jade.tilab.com/> (дата обращения: 02.09.2014).
18. Poslad S., Buckle P., Haddingham R. The FIPA-OS agent platform: Open Source for Open Standards // Proc. of PAAM 2000. – Manchester, UK, 2000. – P. 355–368.
19. Гальперов В.И. Применение многоагентного подхода для разработки программных систем оценивания состояния ЭЭС // Системные исследования в энергетике: Труды молодых ученых ИСЭМ СО РАН. Вып. 44. – Иркутск: ИСЭМ СО РАН, 2014. – С. 165–170.
20. Колосок И.Н., Пальцев А.С. Двухуровневый иерархический алгоритм оценивания состояния ЭЭС и его реализация на основе мультиагентного подхода // ЭНЕРГОСИСТЕМА: управление, конкуренция, образование: Сб. докладов III Междунар. научно-практ. конф. Т. 1. – Екатеринбург, УГТУ-УПИ, 2008. – С. 354–359.

Поступила 10.09.2014 г.

UDC 004.89

DEVELOPMENT OF MULTI-AGENT SYSTEMS FOR DISTRIBUTED SOLUTIONS OF ENERGY PROBLEMS USING AGENT-BASED SCENARIOS

Ludmila V. Massel,

Melentiev Energy Systems Institute of Siberian Branch
of the Russian Academy of Sciences, 130, Lermontov street,
Irkutsk, 664033, Russia. E-mail: massel@isem.sei.irk.ru

Vasilii I. Galperov,

Melentiev Energy Systems Institute of Siberian Branch
of the Russian Academy of Sciences, 130, Lermontov street, Irkutsk,
664033, Russia. E-mail: galperov@gmail.com

On the one hand, the relevance of the research is defined by the evolving concept of the Smart Grid, which important aspect is the usability of the multi-agent technologies. And on the other hand the relevance is defined by a few number of practical approaches to implementation of multi-agent system. Electrical power system function and management nowadays require the development of a big sized model. Such models are not fully observable: there can be data corruption and bad synchronization. As a result, experts can make wrong decisions. To solve this problem we need to create new methods and software for state evaluation.

The main aim of the research is to provide an overview of current agent solutions on IT market; to analyze the cons of the existing systems and to offer an agent scenario based approach for developing multi-agent systems to solve energy problems that will avoid these cons.

Research method. Nowadays the agent approach is not used for developing the systems that solve the problems in electrical energy industry or estimate electrical power system condition in particular. The proposed approach for developing a multi-agent system should allow interaction of agents, created on different program platforms, and also creating new and editing existing algorithms in the system by agent scenarios.

Results. The authors have developed the approach for creating a multi-agent system to estimate the electric power system state with the support of cross-platform interactions and agent scenarios. The approach was tested and a system for estimating electrical power system condition is being developed now.

Key words:

Agent, multi-agent systems, estimation of EPS state, agent-based scenarios, Smart Grid.

The research was partially financially supported by the RFBR grants no. 15–07–01284, 13–07–00140, 14–07–00116, 15–57–04074 Bel_mol_a as well as by the grants of the RAS Presidium program no. 229.

REFERENCES

1. Gamm A.Z., Gerasimov L.N., Golub I.I., Grishin Yu.A., Kolosok I.N. *Otsenivanie sostoyaniya v elektroenergetike* [State estimation in electric power]. Moscow, Nauka Publ., 1983. 302 p.
2. Gamm A.Z. *Statisticheskie metody otsenivaniya sostoyaniya elektroenergeticheskikh sistem* [Statistical methods for state estimation of electric power systems]. Moscow, Nauka Publ., 1976. 220 p.
3. Kobets B.B., Volkova I.O. *Innovatsionnoe razvitie elektroenergetiki na baze kontseptsii Smart Grid* [Innovative development of electric power based on Smart Grid concept]. Moscow, IATs Energiya, 2010. 208 p.
4. Phadke. G. Synchronized Phasor Measurements. A Historical Overview. *IEEE/PES Transmission and Distribution Conference*, 2002, no. 1, pp. 476–479.
5. Galperov V.I., Kolosok I.N., Massel L.V., Paltsev A.S. *Postanovka zadachi razrabotki multiagentnoy sistemy dlya otsenivaniya sostoyaniy EES s uchetoм strukturnoy i funktsionalnoy dekompozitsii* [Development of a multiagent system for power grid state estimation including structural and functional decomposition]. *Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii: Trudy XVIII Baykalskoy Vserossiyskoy konferentsii. Ch. III* [Information technology and mathematical science and management. Proc. of the Baikal

- XVIII All-Russian Conference. P. III]. Irkutsk, ISEM SB RAS, 2013. pp. 231–234.
6. Gorodetskiy V.I. Mnogoagentnye sistemy: sovremennoe sostoyaniye issledovaniy i perspektivy primeneniya [Multi-agent systems: current state and perspectives of research]. *AI News*, 1996, no. 1, pp. 44–59.
7. Gorodetskiy V.I. Mnogoagentnye sistemy: osnovnye svoystva i modeli koordinatsii povedeniya [Multi-agent systems: the basic properties of the model and the coordination of behavior]. *Informatsionnye tekhnologii i vychislitelnye sistemy – Information technology and computer systems*, 1998, no. 1, pp. 22–34.
8. Russell S., Norvig P. *Artificial Intelligence: a Modern Approach*. Translated from English. Moscow, Publishing House «Williams», 2006. 1408 p.
9. Mnogoagentnye sistemy [Multi-agent systems]. Available at: <http://www.aiportal.ru/articles/multiagent-systems/multi-agent-systems.html> (accessed: 02 September 2014).
10. Mnogopodkhodnoe imitatsionnoe modelirovanie [Multimethod Simulation Software]. *AnyLogic*. Available at: <http://www.anylogic.ru/features> (accessed: 02 September 2014).
11. Odell J., Giorgini P., Muller J.P. Agent-Oriented Software Engineering. *4th International Workshop, AOSE 2003*. New York, USA, July 2003. pp. 47–59.
12. Bresciani P., Giorgini P., Henderson-Sellers B., Low G. Agent-Oriented Information Systems II. *6th International Bi-Conference Workshop, AOIS 2004*. Riga, Latvia, June 8, 2004; New York, USA, July 20, 2004. pp. 110–124.
13. Cavedon L., Maamar Z., Martin D., Benatallah B. Extending Web Services Technologies: the Use of Multi-Agent Approaches. *Multiagent Systems, Artificial Societies, and Simulated Organizations*. New York, Springer US, 2004. pp. 29–52.
14. Fartyshev D.A., Chernousova E.S., Chernousov A.V. Podkhod k razrabotke mnogoagentnoy raspredelennoy intellektual'noy informatsionnoy sistemy dlya issledovaniy v energetike [Approach to the development of multi-agent distributed intelligent information systems for research in energy]. *Computational Technologies*, 2008, vol. 13, Special Iss. 1, pp. 108–115.
15. Fartyshev D.A., Chernousov A.V. Metody ispolzovaniya Web-servisov dlya postroeniya vychislitelnoy IT-infrastruktury sistemnykh issledovaniy v energetike [Methods of using Web-services for construction of computing IT infrastructure systems studies in power engineering]. *Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii: Trudy XII Baykalskoy Vserossiyskoy konferentsii. Ch. 2* [Information and Mathematical Technologies in Science and Control. Proc. of Baikal All-Russian XII conference. P. 2]. Irkutsk, ISEM SB RAS, 2007. pp. 46–54.
16. FIPA Abstract Architecture Specification. Available at: <http://www.fipa.org/specs/fipa00001/SC00001L.html> (accessed: 07 September 2014).
17. JAVA Agent DEvelopment Framework. *Jade*. 2015. Available at: <http://jade.tilab.com/> (accessed: 02 September 2014).
18. Poslad S., Buckle P., Hadingham R. The FIPA-OS agent platform: Open Source for Open Standards. *Proc. of PAAM 2000*. Manchester, UK, 2000. pp. 355–368.
19. Galperov V.I. Primenenie mnogoagentnogo podkhoda dlya razrabotki programmnykh sistem otsenivaniya sostoyaniya EES [Application of multi-agent approach to develop software systems of state estimation EPS]. *Sistemnye issledovaniya v energetike: Trudy molodykh uchenykh ISEM SO RAN. Vyp. 44* [System Research in Energy. Proc. of Young Scientists ESI SB RAS. Iss. 44]. Irkutsk, ESI SB RAS, 2014. pp. 165–170.
20. Kolosok I.N., Paltsev A.S. Dvukhurovnevny ierarkhicheskiy algoritm otsenivaniya sostoyaniya EES i ego realizatsiya na osnove multiagentnogo podkhoda [A two-level hierarchical state estimation algorithm EES and its implementation based on multi-agent approach]. *ENERGOSISTEMA: upravlenie, konkurentsia, obrazovanie: Sbornik dokladov III Mezhdunarodnoy nauchno-prakticheskoy konferentsii. T. 1* [POWER SYSTEM: management, competition, education. International scientific-practical conference. Vol. 1]. Ekaterinburg, Ural State Technical University, 2008. pp. 354–359.

Received: 10 September 2014.