

В. И. Городецкий, проф., гл. науч. сотр., gor@iias.spb.su,
О. Л. Бухвалов, мл. науч. сотр., psychoveter@gmail.com,
Санкт-Петербургский институт информатики и автоматизации РАН

Самоорганизующиеся производственные B2B-сети. Часть 2. Архитектура и алгоритмическая поддержка^{1, 2}

Рассматривается сетевая модель ведения распределенного бизнеса, которая является в настоящее время одной из базовых моделей цифровой экономики на уровне управления производством. Даны постановки основных прикладных задач управления производством в концепции B2B-сети, рассмотрено базовое множество задач программно-коммуникационной инфраструктуры, поддерживающей самоорганизующийся стиль работы открытой производственной B2B-сети и распределенную координацию процессов управления производством. Предложены варианты решений перечисленных задач и показана практическая реализуемость концепции B2B-сети производственных предприятий уже в настоящее время.

Ключевые слова: B2B-сети, многоагентная архитектура, самоорганизация, распределенная координация, программно-коммуникационная инфраструктура, планирование, составление расписаний

Введение

Обычно B2B-сетями называют вид экономического и информационного взаимодействия предприятий, компаний или организаций иного типа (например, виртуальных предприятий, распределенных производств, распределенных транспортных компаний и др.), имеющий целью координацию распределенного выполнения общих бизнес-процессов *в интересах самого бизнеса*. В данной работе рассматриваются самоорганизующиеся производственные B2B-сети.

В первой части работы проанализировано текущее состояние исследований и разработок в этой области, сформулированы ее ключевые проблемы и вызовы, предложена концепция реализации парадигмы производственной B2B-сети как самоорганизующейся открытой сети предприятий, построенной на основе метафоры многоагентных систем. В ней были содержательно сформулированы базовые прикладные задачи, которые должны решаться в сети для реализации функций распределенного управления производственными процессами, а также функции программно-коммуникационной инфраструктуры, предназначенной для поддержки распределенного взаимодействия компонент управления производством, обеспечения ее открытости и реализации самоорганизующегося стиля функционирования. В данной части работы рассматривается алгоритмическая поддержка перечисленных задач (разделы 1 и 2, а также общая

архитектура организации программного обеспечения B2B производственной сети (раздел 3). В заключении формулируются общие выводы по результатам работы и подчеркивается важная роль концепции B2B-сети как парадигмы организации распределенного производства в ближайшем будущем.

Основные прикладные задачи, решаемые в узлах B2B-сети и на сетевом уровне, включают в себя планирование выполнения заказов, составление локальных расписаний исполнения заказов в узлах сети и распределенную координацию локальных расписаний для тех заказов, которые выполняются в нескольких узлах. Содержание прикладных задач и алгоритмы их решения во многом определяют требования к составу задач инфраструктурной компоненты программного обеспечения B2B-сети. Далее конкретизируются постановки названных прикладных задач управления и задач инфраструктурной компоненты, а также даются описания алгоритмов их решения.

1. Базовые алгоритмы для прикладных задач производственной B2B-сети

1.1. Постановка и алгоритм решения задачи планирования выполнения заказов

В постановке этой задачи предполагается, что каждый заказ или его часть ("подзаказ"), подлежащая исполнению в конкретном цехе, заданы частично упорядоченным множеством операций, которые при планировании на уровне цеха не могут подвергаться дальнейшей декомпозиции.

В случае открытой B2B-сети исполнителем любой части заказа может стать любой узел сети, обладающий необходимыми компетенциями и ресурсами. Полагается, что каждый узел (предприятие

¹ Данная работа выполнена в рамках исследований по проекту № 0073-2015-0003 бюджетной тематики СПИИРАН и по проекту № 214 Программы Президиума РАН I.5П.

² Первая часть статьи опубликована в журнале "Мехатроника, автоматизация, управление", 2017, Т. 18, № 11.

сети), анализируя возможность выполнения некоторого заказа или его части, автономно оценивает достаточность своих компетенций и ресурсов, а также возможность их использования в требуемом интервале времени. Предполагается также, что каждый потенциальный исполнитель заказа/подзаказа адекватно оценивает свои возможности по реализации тех или иных функций и является добросовестным участником В2В-сети, а все последующие отклонения от заявленных возможностей, нарушения требований по качеству и срокам исполнения заказов обусловлены случайными возмущениями, которые не зависят от воли участников В2В-сети. Другими словами, в постановке задачи не рассматриваются риски, связанные с возможной недобросовестностью исполнителей заказов.

Каждый узел сети обладает ограниченным набором технологических способностей по производству продуктов и/или по предоставляемому им множеству сервисов. Он также имеет ограниченные свободные ресурсы, которые могут быть привлечены к выполнению заказов в требуемый период времени. Другие предположения постановки задачи планирования заказов таковы:

- заказы могут поступать в любой узел В2В-сети или быть им инициированы. Узел, который инициировал заказ, или узел, в который заказ поступил извне, инициализирует и реализует процесс планирования его исполнения;
- планирование исполнения заказов выполняется в реальном времени, причем на фоне уже существующей загрузки узлов и в динамике поступления заказов;
- узлы сети являются равноправным, т.е. потенциальные заказчики имеют одинаковый доступ ко всем потенциальным исполнителям, а все исполнители могут участвовать в исполнении заказов в соответствии со своими компетенциями и наличными ресурсами;
- планирование выполняется в распределенном варианте, централизация планирования и координации исполнения заказов не используются³;
- инфраструктура В2В-сети должна быть способна выполнять поиск требуемых сервисов и исполнителей с помощью распределенных механизмов белых и желтых страниц [12, 13] (см также далее);
- конфиденциальная информация узлов сети о технологиях, которыми они владеют, о ресурсах, которыми они располагают, и о загруженности их другими заказами должна быть недоступна другим узлам и потому не может использоваться протоколом распределенного планирования;

³ Это требование является достаточно жестким и не всегда адекватно практике. Но оно здесь вводится как особенность постановки задачи. На практике задача может быть проще, если для отдельных подзадач допускается централизованный вариант. Это, например, может относиться к белым и желтым страницам агентской платформы и компонентам единой информационной среды предприятия, если модель В2В-сети масштабируется на уровень отдельного предприятия.

- качество плана выполнения заказа определяется степенью соблюдения директивных сроков его выполнения, а также платой за выполнение заказа, которую предприятие получает от заказчика.

Результатом работы механизма распределенного планирования является назначение заказов или их частей конкретным исполнителям. Заданный частичный порядок на множестве операций заказа нарушаться не может, поскольку он определяется технологической картой исполнения заказа. Заметим, что в процессе оперативного управления распределенным исполнением заказов узлы сети должны быть обеспечены еще некоторой дополнительной информацией, необходимой для поддержания их ситуационной осведомленности в целях временной синхронизации процессов исполнения заказа в соответствии с заданным частичным порядком на множестве его операций. Для каждого заказа указываются также временные рамки его исполнения, т.е. самое раннее допустимое время начала исполнения заказа и директивный срок его исполнения.

Для решения описанной задачи распределенного планирования заказов используется стандартный рыночный механизм самоорганизации [1, 2], в основе которого лежит модель аукциона. В этой модели в конкретной сессии аукциона любой участник рынка играет либо роль *покупателя* товара (в конкретных приложениях товаром могут быть задачи, станки производственной системы, сервисы, ресурсы и т.п.), либо роль *продавца* (владельца ресурса), причем оба они действуют только в своих интересах. Вторым важным компонентом модели аукциона является явно описанный механизм установления соответствия ("мэтчинга") владельцев и потребителей ресурсов, представляющих интерес друг для друга в рамках возможной "сделки".

В сессии аукциона узлы В2В-сети взаимодействуют локально с помощью обмена сообщениями, которые содержат *предложения, сделки, обязательства и платежи* за ресурсы. Аукционер, выступающий от имени заказчика (в этой роли выступает узел сети, ответственный за планирование заказа), готов заплатить исполнителю заказа или его части определенную цену в пределах некоторого бюджета, и его цель — заплатить как можно меньше. Узлы сети, обладающие необходимыми компетенциями и свободными ресурсами для исполнения заказа, которые заинтересованы в его получении, наоборот, имеют целью максимизировать свой доход. Они решают сами, участвовать ли им в торгах за заказ и на какую минимальную плату за его исполнение они согласны.

В описанной стратегии самоорганизации каждый участник аукциона имеет свою максимальную/минимальную цену, которая неизвестна другим узлам, и при этом покупатель не может платить больше установленной им максимальной цены, а продавец не может продавать ресурс за меньшую цену, чем им выбрана. Торги строятся

так, чтобы в их процессе обе цены эволюционировали по направлению к рыночной цене [3–6].

Таким образом, общая идея механизма распределенного планирования выполнения множества заказов и их частей в В2В-сети сводится к проведению множества *локальных аукционов* по множеству заказов/подзаказов на динамически формируемом *локальном множестве узлов* сети, которые рассматриваются как потенциальные исполнители соответствующих заказов/подзаказов. Алгоритмическая сторона этой модели формализуется в терминах соответствующих *протоколов*. В экспериментальной разработке для этих целей использовался простейший частный случай аукциона, который реализуется протоколом контрактных сетей [7]. Его описание дается в следующем подразделе. В области многоагентных систем он имеет стандартную реализацию [8].

1.2. Постановка и алгоритм решения задачи составления расписаний выполнения заказов

Требования, изложенные в предыдущем подразделе применительно к задаче планирования исполнения заказов, по большей части относятся также и к задаче составления расписания. Важным ограничением и в этом случае остается требование конфиденциальности информации, касающейся технологических, ресурсных и других аспектов узлов В2В-сети. Очевидно, что это ограничение делает неприменимыми все существующие централизованные методы построения расписаний.

Рассмотрим содержание задачи составления расписаний выполнения заказов, назначенных узлам сети на этапе планирования. Этот алгоритм, как он реализуется авторами в экспериментальных разработках по данной тематике [2], можно разделить на две части. Одна из них — это та часть алгоритма, которая выполняется "внутри" каждого узла сети. Она имеет целью построение локального расписания выполнения заказов/частей заказов, которые конкретный узел должен выполнить в соответствии с результатами решения задачи планирования. Эта часть алгоритма, а также те конфиденциальные данные, которые используются им, "не видны" извне. Вторая часть алгоритма имеет целью координацию локальных расписаний узлов в целях обеспечения ограничений, наложенных на заказы. Эти ограничения задают частичный порядок на множестве его частей, исполняемых разными узлами, а также задают допустимые временные рамки выполнения заказа в целом.

Алгоритм координации описывается протоколом взаимодействия узлов. Он реализуется путем обмена сообщениями, имеющими целью согласование локальных расписаний таким образом, чтобы временные ограничения, наложенные на заказ в целом, были бы, по возможности, выполнены. Напомним, что ограничения на частичный порядок, заданный на множестве частей заказа, должен быть выполнен безусловно. Не может также нару-

шаться ограничение на самое раннее время начала выполнения заказа. Что касается директивного срока выполнения заказа, то задержки с выполнением заказа являются объектом минимизации по некоторому показателю, представленному, например, в терминах функции штрафа. Далее содержательно описываются локальные алгоритмы составления расписаний выполнения заказов в узлах сети, а также алгоритм координации локальных расписаний на уровне В2В-сети.

Локальный алгоритм составления расписаний. Рассмотрим сначала формальную постановку задачи составления локального расписания узла В2В-сети. Пусть $[T_0, T_k]$ — интервал времени, для которого составляется расписание работы узла. Обозначим множество заказов узла, подлежащих исполнению в заданном горизонте планирования, $O = \{O_j\}_{j=1}^N$, где O_j — заказ с номером j , а N — общее число заказов, для которых составляются расписания. Напомним, что каждый заказ множества O представлен частично упорядоченным множеством операций. Каждому заказу O_j поставлено в соответствие значение времени $T_0^{(j)}$, раньше которого его исполнение начато быть не может. Заказы, исполнение которых уже было начато до момента времени T_0 , в множество O не включаются. Все остальные заказы участвуют в процессе составления локального расписания на равных правах. Если некоторый ресурс R_i на момент времени T_0 занят выполнением некоторого заказа, то в новом расписании он может быть использован только после того, как освободится. Формально это учитывается индивидуальной диаграммой временной доступности ресурса.

Каждой операции заказа поставлены в соответствие требования к ресурсам, которые необходимы для ее выполнения. В производственных сетях к ресурсам относятся, например, рабочие той или иной квалификации, станочное оборудование, на котором операция может исполняться, материалы, потребные по технологии, и др. Вся информация о ресурсах предприятия обычно бывает представлена в соответствующей справочно-информационной системе, которая является компонентой единого информационного пространства узла сети и которая доступна алгоритмам составления расписаний.

Предполагается, что расписание строится так, чтобы оптимизировать локальный показатель качества или несколько таких показателей в более общем случае. Обычно используются следующие показатели:

- *объем товарной продукции*, которая должна быть произведена в течение календарного месяца; эта величина определяется суммой нормо-часов всех заказов, *завершаемых* в течение текущего месяца;
- *объем валовой продукции*, произведенной в течение календарного месяца; эта величина определяется суммой нормо-часов всех *запланированных* операций.

Достижение заданных объемов товарной и валовой продукции рассматривается в качестве критериев эффективности искомого расписания.

Кроме того, поскольку очень часто бывает невозможно выполнить временные ограничения на сроки выполнения заказов, то в качестве еще одного показателя качества расписания выступает значение некоторой функции штрафа за нарушение этих сроков.

Разработанный алгоритм составления локального расписания исполнения заказов узла строится как итеративная процедура. В ней на каждой итерации:

а) проводится выбор очередной операции, для которой определяется интервал времени выполнения, и

б) находится допустимое время начала ее исполнения и ресурсы, которые будут использованы при ее выполнении.

Очередная операция выбирается из множества операций, все предшественники которых по отношению порядка операций в заказе уже включены в расписание. Выбор очередной операции из этого множества выполняется с помощью некоторой *политики*, содержащей эвристические правила. Выбор конкретного ресурса узла и времени начала исполнения операции выполняется с помощью протокола контрактных сетей [7, 8]. Его стандартная AUML-диаграмма представлена на рис. 1. Детали алгоритма составления локального расписания и разработанные программные версии их реализации описаны в работах [1, 9].

Алгоритм распределенной координации локальных расписаний узлов. В этом алгоритме рассматривается множество узлов B2B-сети $\{Dep_i\}_{i=1}^{N_{Dep}}$ (*Dep* — от англ. *Department*), совместно выполняющих заказы множества $O = \{O_j\}_{j=1}^N$, из которых, по крайней

мере, некоторое подмножество таково, что в их исполнении участвуют несколько узлов.

Каждому заказу, назначенному для выполнения подмножества узлов B2B-сети, ставятся в соответствие два времени: $\{T_0^{(j)}\}_{j=1}^{N_{Order}}$ — самое раннее время, когда выполнение заказа $O = \{O_j\}_{j=1}^N$ может

быть начато, и $\{T_k^{(j)}\}_{j=1}^{N_{Order}}$ — самый поздний срок, к которому он должен быть выполнен. Подчеркнем, что эти ограничения относятся к заказу в целом, а не к его частям. Они задают интервал календарного времени, в течение которого заказ должен быть выполнен в соответствии с контрактом.

Алгоритм распределенной координации расписаний должен:

1) обеспечить синхронизацию выполнения различных частей заказа, исполняемого несколькими узлами, таким образом, чтобы безусловно выполнялись отношения частичного порядка, заданные на их множестве;

2) по возможности минимизировать задержки выполнения заказов по отношению к директивным временам их исполнения. Действительно, интервал времени исполнения заказа O_j в соответствии с контрактом должен укладываться в интервал $[\{T_0^{(j)}\}_{j=1}^{N_{Order}}, \{T_k^{(j)}\}_{j=1}^{N_{Order}}]$, но каждый локальный алгоритм составления расписаний не имеет информации об этом. По этой причине ответственность за выполнение данного ограничения возлагается на алгоритм координации расписаний.

Базовую идею алгоритма иллюстрирует рис. 2. Дадим его краткое описание. Алгоритм распределенной координации локальных расписаний узлов B2B-сети строится на основе обмена метаданными

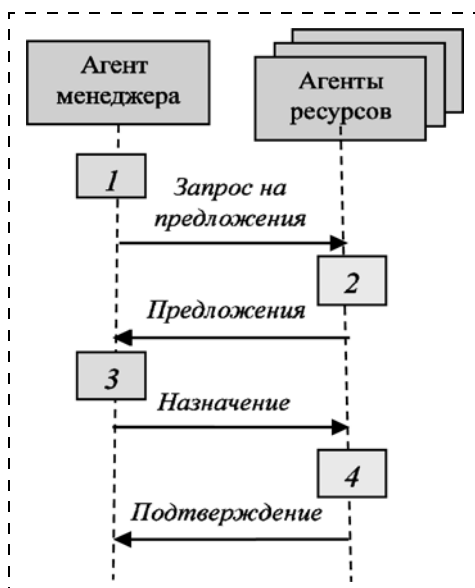


Рис. 1. AUML-диаграмма CNP-протокола

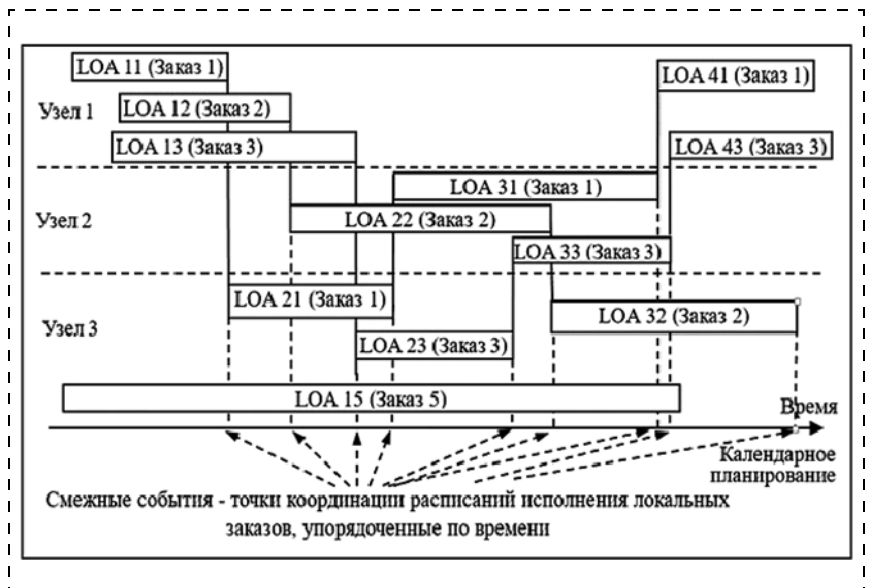


Рис. 2. Пояснение базовой идеи алгоритма распределенной координации. В примере общее число заказов — 5, число узлов — 3

между парой ее узлов, которые выполняют две смежные (по отношению порядка) части одного и того же заказа. Запуск алгоритма распределенной координации (*протокола координации*) осуществляется так называемыми *смежными событиями*⁴. Каждое такое событие генерируется тогда, когда заканчивается составление расписания исполнения части заказа, выполняемой в некотором узле, а продолжение выполнения этого заказа есть функция уже другого узла (или узлов), который ответствен за выполнение смежной (непосредственно следующей) операции заказа для той из операций, которая была инициатором генерации смежного события.

Другое важное понятие алгоритма координации — это понятие *маркера фронта работ*. Обозначим операцию r заказа O_j символом $a_r(O_j)$ (от англ. *activity*). Напомним, что в каждом узле сети в соответствии с локальным расписанием параллельно обычно выполняются несколько операций одного или нескольких заказов. Множество таких операций в локальном расписании на момент времени t называется *фронтом работ узла* в этот момент времени. Соответственно, *маркером фронта работ* называется метка $M(a_r(O_j), t) = M_{r,j}(t)$, которая ставится в соответствие каждой работе $a_r(O_j)$, принадлежащей в момент времени t фронту работ узла.

Понятие смежного события является базовым понятием алгоритма распределенной координации, поскольку оно реализует его операционную семантику. Каждый раз, когда генерируется смежное событие, управление процессом исполнения заказа передается алгоритму координации расписаний. Координация выполняется путем переговоров двух узлов: узла сети, в котором завершилось выполнение порученной ему части заказа (обозначим его N^-), и узла, в котором выполнение заказа должно быть продолжено (обозначим его N^+).

Смежное событие $E_{r,j}(N^-, N^+, t)$, сгенерированное узлом сети N^- , содержит в качестве аргумента имя узла N^+ , которому должна быть сообщена информация об окончании выполнения соответствующей части заказа O_j в узле N^- . Узел N^+ должен продолжить его выполнение. Заметим, что в общем случае таких узлов может быть несколько, и тогда смежное событие содержит в качестве аргументов имена всех таких узлов. В частном случае последователя для сгенерированного смежного события может и не быть, если операция, инициировавшая смежное событие, является последней в заказе O_j . Таким образом, с каждым смежным событием $E_{r,j}(N^-, N^+, t)$ связано два факта, существенных для того узла (узлов), в котором (которых) должно продолжаться выполнение заказа:

(а) окончание предшествующей операции и

⁴ *Смежные события*, наряду с *маркерами*, о которых речь идет чуть ниже, являются компонентами информации, которой обмениваются агенты узлов для поддержания их ситуационной осведомленности, о которой упоминалось ранее.

(б) разрешение включить во фронт работ узла-получателя смежного события N^+ последовательной работы O_j , законченной в узле сети N^- .

Локальный цикл составления расписания продолжается до тех пор, пока имеется хотя бы одна операция, помеченная маркером. В случае, когда таких операций нет, локальный цикл работы алгоритма переходит в режим ожидания и стартует тогда, когда в узле появляются операции, помеченные маркером.

Другими словами, алгоритм организован таким образом, что глобальный фронт планирования (объединение фронтов по множеству узлов сети) продвигается в нем и по временной шкале, и по структурированному множеству операций заказов сети. Когда заканчивается планирование некоторого заказа, в конечном узле выполняются также некоторые заключительные операции.

2. Базовые алгоритмы инфраструктурных задач производственной В2В-сети

Напомним, что под *инфраструктурными задачами* понимается множество задач, которые в В2В-сети необходимо решать для поддержки функций коммуникаций, поиска объектов и сервисов, организации взаимодействия узлов (предприятий) В2В-сети, и другие аналогичные задачи, которые непосредственно не относятся к функциям приложений. Далее кратко рассматриваются постановки и алгоритмы решения этих задач. Заметим, что задача обеспечения безопасности сети здесь не рассматривается.

2.1. Обеспечение коммуникационного сервиса

Физической основой этого сервиса является сеть Интернет. Коммуникационный сервис должен *обеспечивать отправку сообщений* как автономным программным сущностям того же узла, так и сущностям, установленным в других узлах сети, для которых текущий узел является либо адресатом сообщения, либо промежуточным узлом для поиска нужного адресата с помощью некоторого протокола маршрутизации. Этот сервис должен *предоставлять каналы связи* и обеспечивать *адресацию сообщений*. Ввиду предположения об открытости сети, в которой топология и состав узлов могут меняться со временем, эта адресация должна выполняться по протоколу, отличному от протокола *TCP/IP*. В случае, если для адресации сообщений или поиска сервиса по его имени используется р2р-протокол, коммуникационную компоненту, обеспечивающую сервис предоставления каналов связи и адресации сообщений, принято называть *пиром* (от англ. *peer* — одноранговый, равноправный; этот термин подчеркивает горизонтальный характер связи, в отличие от иерархического). Пир должен также исполнять *механизм управления списком своих контактов* в том случае, если в сети изменяется состав узлов. Эта задача решается при из-

менении топологии сети, что может происходить как за счет выхода узлов из сети и входа в нее новых узлов, так и за счет намеренной реконфигурации сети с целью оптимизировать ее топологию⁵.

2.2. Обеспечение открытости сети

Открытость сети обеспечивается, с одной стороны, множеством стандартных протоколов, с помощью которых произвольное предприятие может войти в сеть, объявить о своих сервисах, получить информацию о соседях и их сервисах и стать таким образом равноправным узлом сети. Вместе с тем, открытость сети обеспечивается стандартными протоколами, которым следуют узлы сети, намеревающиеся покинуть сеть. При этом инфраструктура должна обеспечить поддержку связности сети, маршрутизацию сообщений при модификации топологии сети и зафиксировать изменение состава сервисов, доступных в ней. Такая постановка задачи рассматривается в литературе по открытым сетям. Для ее решения могут быть использованы различные принципы.

Один из вариантов — это использование технологии *JXTA* [10], которая является *Java*-реализацией стандарта, разработанного *UPnP*-форумом [11]. Однако к настоящему времени компания *Ogacle*, владелец технологии *JXTA*, остановила работы по ней и поддержку разработок. По этой причине судьба этой технологии в настоящее время неясна, как неясны и перспективы ее использования в самоорганизующихся *B2B*-сетях.

Другой, более поздний вариант возможной реализации механизма открытости сети базируется на рабочем стандарте *FIPA* [12]. Он предлагает использовать распределенную *p2p*-платформу агентов, обеспечивающую типовой механизм *p2p*-взаимодействия в сети, который обеспечивает "полное разъединение связи агента с пиром от прикладного уровня, позволяя их независимую модификацию" [12, 13]. При этом распределенная агентская платформа реализуется как оверлейная (виртуальная) сеть, установленная поверх коммуникационного уровня, представляемого сетью пиров. Аналогично прикладной уровень, образуемый узлами *B2B*-сети, реализуется в этой концепции как оверлейная сеть, установленная, в свою очередь, поверх оверлейной сети экземпляров распределенной агентской платформы. Заметим, что коммуникационный уровень, если он организуется для работы в *p2p*-стиле, также строится как оверлейная сеть, установленная поверх *TCP/IP*-уровня.

Однако существуют и другие возможности поддержки открытости *B2B*-сети, базирующиеся на обмене сообщениями, например акторные систе-

мы, различные реализации стандарта *JMS* (от англ. *Java Messaging System*) и т.п. В частности, библиотека акторов *Akka* в последнее время стала очень популярной, так как предлагает технологию, позволяющую прозрачно разрабатывать распределенные системы. В настоящее время активно разрабатывается модуль этой библиотеки *Akka Cluster*, предоставляющий механизмы для открытых сетей. Имеются также случаи успешного применения *JMS* в мобильных *ad-hoc* сетях [14]. В качестве сервиса распределенных желтых страниц здесь может быть использована технология *Apache ZooKeeper*. Она централизованная, но центральный узел (лидер) выбирается из узлов, входящих в сеть. Если лидер *ZooKeeper* выйдет из строя, то выбирается новый лидер. Этот проект развивается уже давно и используется, например, в решении *Storm* (кластерная технология для обработки потоков данных в реальном времени), созданном компанией *Twitter*.

В итоге анализа достоинств и недостатков описанных вариантов обеспечения открытости сети, а также с учетом имеющегося собственного опыта в области разработки распределенной *p2p*-агентской платформы в данной работе выбор остановлен на использовании библиотеки акторов *Akka* для реализации *p2p*-агентской платформы, близкой к рабочему стандарту *FIPA* [12].

Заметим, что в последнее время появились и другие возможности по реализации *p2p*-концепции программно-коммуникационной инфраструктуры для реализации открытых *ad-hoc* сетей. Но они появились недавно, поэтому их достоинства и недостатки еще предстоит оценить.

2.3. Обеспечение информационной совместимости предприятий

Эта задача, решаемая в интересах интеграции предприятий в *B2B*-сеть, является одной из наиболее важных для поддержки возможности участия нескольких предприятий сети в распределенном выполнении общих бизнес-процессов и предусматривает реализацию:

- стандартных интерфейсов и сетевых протоколов, поддерживающих обмен сообщениями. В этом контексте нужно выбрать технологии обеспечения связи, допускающие интеграцию с существующими корпоративными информационными системами (КИС);
- стандартного языка обмена сообщениями. Сообщения, помимо контента, содержат также определенные "перформативы", т.е. коды (слова), управляющие протоколами распределенного взаимодействия. Необходимо описать множество управляющих перформативов для разработанных протоколов, а также разработать гибкую систему управления протоколами (в узлах), позволяющую расширять множество протоколов, доступных в сети. За основу можно взять язык *ACL*, предложенный в качестве стандарта в многоагентных системах;

⁵ В настоящее время эти задачи решаются в научном направлении, которое принято называть *Software-Defined Networking* (сети, существующие на уровне программного обеспечения, виртуальные сети, *SDN*-сети). Частным случаем *SDN*-сетей являются так называемые оверлейные сети.

- стандартной формы представления информации. Необходимо выбрать единый язык описания содержимого сообщений (например, можно взять *BPEL* [15]), разработать онтологию (ее метауровень) для описания требований со стороны заказов к ресурсам, а также свойств заказов.

2.4. Поддержка стандартных протоколов взаимодействия узлов сети

Задача построения стандартных протоколов взаимодействия узлов сети в различных случаях использования (англ. *use cases*) и в конкретных сценариях их реализации имеет целью обеспечение единообразия протоколов взаимодействия узлов сети при решении содержательных задач. Примером этих задач является задача распределенного планирования заказа, поступившего в сеть через некоторый узел. Для решения задач планирования в данной работе предполагается использовать протокол контрактных сетей [16] или другой протокол аукциона. Еще одним примером протокола взаимодействия узлов В2В-сети может быть протокол распределенной координации локальных расписаний работы множества узлов сети при совместном выполнении ими множества бизнес-процессов.

Заметим, что все перечисленные здесь протоколы, как будет показано далее, реализуют *механизмы самоорганизации*.

3. Архитектура и варианты программной реализации В2В-сети

3.1. Многоагентные системы, самоорганизация и В2В-сети

Технология самоорганизующихся систем рассматривается в настоящее время как единственная информационная технология, которая в состоянии предложить адекватные методы, архитектуры и инструментальные средства поддержки для разработки и программной реализации наиболее сложных современных приложений, для которых характерны открытость, большая размерность, автономность составляющих ее подсистем, их сетевая организация, а также мобильность. Именно такие свойства характерны для В2В-сетей. Эффективная реализация В2В-сетей в иной архитектурной парадигме в настоящее время не представляется возможной.

В соответствии с общепринятым определением *самоорганизация* есть динамический адаптивный процесс, в котором структура и поведение системы возникают и поддерживаются только на основе локальных взаимодействий ее компонентов (обмена информацией, сообщениями) без какого-либо явного внешнего управления. Более детально самоорганизующиеся системы характеризуются следующим набором свойств [17]:

- автономность, т.е. отсутствие какого-либо управления извне;

- глобальный порядок (организация, структура), который возникает в системе только за счет локальных взаимодействий ее компонентов;
- эмерджентные свойства, которые не выводятся из наблюдения индивидуального поведения компонент, а наблюдаются только на метауровне;
- избыточность компонентов и их взаимодействий, а потому нечувствительность системы к локальным сбоям, что обеспечивает ее отказоустойчивость;
- адаптивность, т.е. способность системы изменять свое поведение и устойчивое состояние при изменении ее структуры и внешней среды;
- сложность, обусловленная тем, что самоорганизующиеся системы обычно состоят из большого числа автономных компонентов;
- простота правил взаимодействия компонентов системы, реализующих сложное эмерджентное поведение системы в целом.

Описанные свойства самоорганизующихся систем являются во многом аналогичными базовым свойствам В2В-сети как объекта управления, а вместе с тем, они отвечают требованиям к желаемому поведению этого объекта.

Большинство программных прототипов самоорганизующихся систем, разработанных к настоящему времени, используют архитектуру многоагентных систем (МАС), и это не случайно: в настоящее время парадигма МАС является единственной парадигмой концептуализации систем, которая имеет все необходимые средства для описания моделей автономных открытых систем сетевой структуры и большой размерности. Более того, основное развитие моделей самоорганизации и методов их программной реализации сейчас происходит именно на основе технологии МАС. Поэтому в данной работе теоретический и технологический базисы построения В2В-сети основываются на модели многоагентной самоорганизации. Покажем, что алгоритмы решения основных прикладных задач, представленных в разделах 1 и 2, используют механизмы самоорганизации.

Действительно, для решения задачи *планирования выполнения заказов*, динамически поступающих в сеть, в работе предлагается использовать *рыночный механизм* самоорганизации [18, 19]. В его основе лежит модель аукциона, частным случаем которого является протокол контрактных сетей. Он и алгоритм распределенной координации используют только локальную информацию и локальные взаимодействия. Напомним, что в этой модели отсутствует вмешательство извне, и все узлы В2В-сети самостоятельно решают, участвовать ли им в торгах в роли покупателя или продавца в транзакциях покупки/продажи некоторого количества того или иного ресурса. При этом в модели аукциона узлы сети устанавливают свои цены, базируясь целиком на *локальной информации*, хотя *косвенно* они получают также информацию об общем количестве ресурса и о потребности в нем *через предложение* конкретной цены.

Таким образом, общая идея самоорганизации процессов распределенного планирования совместного выполнения множества заказов в В2В-сети сводится к проведению множества *локальных аукционов* по множеству подзаказов на динамически формируемом *локальном множестве узлов* сети, которые рассматриваются как потенциальные исполнители соответствующих подзаказов.

Вторая прикладная задача, рассмотренная в разделе 1, а именно задача *составления распределенного расписания* выполнения множества бизнес-процессов, является одной из самых трудных и вычислительно сложных задач управления в В2В-сетях, она также решается на основе самоорганизации. В настоящее время в литературе имеется только одна постановка такой задачи [2], в которой используется механизм самоорганизации на основе волнового механизма взаимных уступок. В данной работе его идеи использованы как базис для построения механизмов распределенной координации расписаний.

Инфраструктурная компонента В2В-сети, как и всякая р2р-сеть, по сути, тоже функционирует на принципах самоорганизации. Описанные в разделе 2 принципы построения желтых и белых страниц и механизмы поиска сервисов и узлов, владеющих искомым сервисом, в разрабатываемой системе предполагается решать на основе локальных взаимодействий узлов сети и р2р-протоколов обмена сообщениями. Заметим, что хорошо известные аналоги р2р-систем, такие как *T-map протокол* [20], *Freenet* [21] и др., классифицируются именно как алгоритмы самоорганизации [18].

3.2. Архитектура В2В-сети и многоагентные системы

Требования, сформулированные к В2В-сетям, а именно открытость, присутствие динамики структуры и отсутствие централизованных регулирующих механизмов, налагают ограничения на возможную архитектуру В2В-сети. Децентрализация управления обычно рассматривается двояко, а именно с точки зрения управления данными (состояниями) и с точки зрения координации процессов. Для организации децентрализованной работы с данными существуют, по крайней мере, два варианта. Первый из них основан на механизме выбора лидера (англ. *leader election*), когда система с помощью специального алгоритма выбирает точку координации. Примером здесь может быть система *Apache ZooKeeper*, реализующая координацию состояний, представленных парами *ключ-значение* (англ. *key-value*) в распределенной среде. Здесь лидер управляет операциями изменения состояний. Этот подход имеет заметный недостаток, а именно отсутствие постоянной доступности системы. В нем, если вдруг узел-лидер выпадает из системы, запускается протокол выбора нового лидера, а это приводит к задержкам в работе системы. Второй вариант не предполагает выбора центра координации, и в нем

эти операции либо используют распределенные хэш-таблицы (так, например, реализована распределенная база данных *Cassandra*), либо реализуются с помощью протоколов координации состояний.

Другая проблема, которая возникает при создании распределенных/децентрализованных систем, обусловлена выбором способа взаимодействия узлов системы. К настоящему времени предложены два способа взаимодействия узлов: обмен сообщениями (англ. *messaging*) и использование досок объявлений (англ. *blackboard*). К последнему способу относятся также все способы организации взаимодействия через базы данных. Недостатком подхода с использованием досок объявлений является то, что она оказывается узким местом в том смысле, что отказы в ее работе приводят к отказу работы всей системы. Вместе с тем, оповещение разных узлов системы об изменении данных приходится организовывать либо на основе регулярных запросов в целях проверки изменения состояния (поллинга), либо с помощью триггеров, срабатывающих при изменении состояний узлов, что снова нагружает доску объявлений.

Взаимодействие на основе сообщений в последние годы становится основным трендом при разработке распределенных систем, поскольку позволяет добиться следующих свойств:

- прозрачность размещения (англ. *location transparency*). Это значит, что подобная система может работать на 10 узлах (или процессорах) так же, как и на одном узле, без дополнительных доработок;
- реактивность, т.е. компоненты системы реагируют сразу при поступлении сообщения, что позволяет добиться более низкого времени отклика системы;
- удобство балансировки (системы, построенные на основе сообщений, всегда проще балансируются).

Подход, использующий абстракцию сообщения на уровне парадигмы, в информационных технологиях представлен многоагентными системами и моделью акторов.

Моделью акторов называют средство организации вычислений, в котором логической единицей является сущность, именуемая актором, способная:

- а) отправить сообщение другому актору;
- б) создать другого актора;
- в) выбрать поведение для обработки очередного сообщения (поведение подразумевает исполнение некоторого кода в качестве реакции на сообщение).

При этом акторы исполняются параллельно, и каждый актор может одновременно обрабатывать только *одно сообщение*. Модель акторов со столь простыми правилами поведения оказалась, тем не менее, очень мощным средством для разработки *параллельных и распределенных* программ. В случае многоагентных систем такой единой модели представления технологии нет. Заметим, что некоторые авторы в понятие МАС включают и те системы, которые построены только на модели акторов.

Самой известной попыткой стандартизовать МАС являются спецификации FIPA. Они добавляются к коммуникационному уровню (*сущность — общение*):

- язык взаимодействия агентов *ACL* (англ. *Agent Communication Language*);
- средства описания содержания сообщений (англ. *Content Language*), в основном базирующиеся на логических языках, в частности на языках модальных и темпоральных логик;
- формальное понятие протокола взаимодействия;
- стандартные базовые протоколы (например, *CNP-protocol* и др.).

FIPA формализует также ряд важных сетевых сервисов, а именно сервисы *желтых страниц* (англ. *directory facilitator*), сервисы *белых страниц* (англ. *naming service*) и сервисы *онтологий* (англ. *ontology services*). Важно отметить, что модель FIPA использует также распределенную структуру *peer-to-peer* платформы [11, 12], в которой посредством распределенных алгоритмов (например, *Gossip*) удается установить так называемое *оверлейное* пространство взаимодействия агентов, в котором агенты могут прозрачно взаимодействовать друг с другом, "не замечая" того, что они физически распределены. Таким образом, платформа, удовлетворяющая базовым требованиям спецификации FIPA (англ. *FIPA-compliant*), позволяет организовать распределенную *peer-to-peer* среду работы агентов, обеспечивая распределенные службы каталогов (желтые страницы), сервис имен, и другие сервисы. Пример ее программной реализации дан в работе [13].

Таким образом, модель МАС как концепция построения B2B-сети позволяет удовлетворить всем требованиям к свойствам и возможностям B2B-сетей, которые ранее обсуждались в разделе 1 части 1 данной работы. Действительно, в основу МАС-парадигмы положены такие принципы функционирования, как (а) взаимодействие и коммуникации компонент, (б) открытость агентской платформы, поддерживающей взаимодействие компонент, и (в) стандартизация базовых протоколов взаимодействия. Эти принципы напрямую соответствуют требованиям со стороны B2B-сетей.

Многоагентный подход в этой части предоставляет достаточно хорошие возможности, которые обусловлены тем фактом, что в МАС основная ставка делается на использование *онтологий*, которые являются одним из уже классических средств для представления знаний (в том числе, метainформации о структуре данных). По сути, схемы баз данных и модели классов *UML* можно считать частными, упрощенными способами описания онтологий. Для работы с моделями онтологий разработано множество инструментов, позволяющих оперировать знаниями, в частности, извлекать знания из окружающего мира, создавать базы знаний, проводить рассуждения в терминах онтологий и др.

Более того, в теории онтологий развиваются такие направления, как выравнивание онтологий

(установление соответствия между несколькими различными онтологиями (англ. *ontology alignment* или *ontology matching* [22])) и онтологические трансформации [23]. Они особенно важны в рамках задачи обеспечения информационной совместимости узлов B2B-сети. Заметим, что метаданные сообщения, представленного на языке *ACL*, содержат в себе имя онтологии, в терминах которой представлено содержимое сообщения. В среде агентов при этом может существовать сервис онтологий. Это обычно специальный агент, предоставляющий информацию об элементах онтологии, а также позволяющий трансформировать сообщения из одной онтологии в другую. Таким образом, многоагентный подход удовлетворяет также и этому требованию к возможностям архитектуры B2B-сети, сформулированному в разделе 1.

Проведенный анализ показывает, что в настоящее время парадигма МАС как концепция построения модели B2B-сети является предпочтительной парадигмой, которая имеет все необходимые средства для выполнения требований к самоорганизующимся системам.

Заключение

В работе рассмотрена сетевая концепция ведения распределенного бизнеса, которую принято называть B2B-сетью, применительно к производственным сетям предприятий. Эта концепция развития промышленного производства рассматривается экономически развитыми странами как уже ближайшая перспектива интеграции предприятий в единое производственное пространство, например, в масштабе Европейского союза.

Производственная B2B-сеть обычно понимается как территориально распределенная сеть предприятий, каждое из которых обладает определенными компетенциями, технологиями и ресурсами. В этой сети полагается, что технологические возможности узлов сети взаимно дополняют друг друга, так что предприятия сети заинтересованы в кооперации, в особенности при производстве сложной продукции. Важно отметить, что концепция производственной B2B-сети может масштабироваться как на уровень множества производственных предприятий, так и на уровень множества подразделений отдельного предприятия. Следует также заметить, что сетевая организация производства является одной из моделей, на которые ориентируется современная цифровая экономика.

Естественно, что новые концепции и модели всегда ставят и новые проблемы. В данной работе проанализированы задачи, которые необходимо решать для того, чтобы концепция сетевой организации производства могла быть реализована на практике.

Первый результат данной работы состоит в том, что в ней сформулирован и проанализирован базовый комплекс задач, которые должна решать система управления производством, организованным в B2B-сеть. С алгоритмической точки зрения

наиболее сложными задачами управления функционирования В2В-сети являются задачи планирования выполнения распределенных бизнес-процессов и составления скоординированных расписаний исполнения планов при условии, что в выполнении одного бизнес-процесса может быть задействовано несколько узлов сети в последовательно-параллельном варианте. Работу этих узлов необходимо координировать как при планировании, так и при составлении расписаний. При этом обычно требуется обеспечить сохранение конфиденциальной информации узлов об их технологических возможностях, наличных ресурсах и их текущей и планируемой загрузке. Опыт участия авторов данной работы в практической реализации отдельных базовых компонентов системы управления сетевым производством отражен в ряде практических разработок по методам распределенного планирования и составления расписаний, описанным в работах [1, 2, 9].

Другой результат работы состоит в том, что в ней проведен анализ архитектуры и базовых задач программно-коммуникационной среды (инфраструктуры), которая преобразует множество отдельных предприятий в единый объект сетевой структуры. Опыт авторов в части практической разработки прототипов подобной инфраструктуры представлен, например, в работе [13].

Еще один результат данной работы — это конкретные варианты решений перечисленных выше задач, которые показывают практическую реализуемость концепции открытой самоорганизующейся В2В-сети производственных предприятий уже в настоящее время.

Разработка программного прототипа В2В-сети с совместным решением всех описанных задач планируется в качестве дальнейших исследований и разработок по теме В2В-сетей.

Список литературы

1. Бухвалов О. Л., Городецкий В. И., Карсаев О. В., Кудрявцев Г. И., Самойлов В. В. Производственная логистика: Стратегическое планирование, прогнозирование и управление конфликтами // Известия Южного федерального университета, 2012. № 3. С. 209—218.
2. Bukhvalov O., Gorodetsky V., Karsaev O., Koudryavtsev G., Samoylov V. Privacy-Preserved Distributed Coordination of Production Scheduling in B2B Networks: A Multi-agent Approach // Proc. 7th IFAC Conference on Manufacturing Modeling, Management, and Control. 2013. Материалы конференции. Vol. 7. Part 1. P. 2122—2127.
3. Chavez A., Maes P. Kasbah: An Agent Marketplace for Buying and Selling Goods // Proc. First Intern. Conf. "The Practical Application of Intelligent Agents and Multi-Agent Technology", London, UK. 1996. P. 75—90.
4. Fisher K., Mueller J. P., Heimig I., Scheer A.-W. Intelligent Agents in Virtual Enterprises // Proc. First International. Conf. "The

Practical Application of Intelligent Agents and Multi-Agent Technology". London, UK. 1996. P. 205—224.

5. Jennings N., Paratin P., Jonson M. Using Intelligent Agents to Manage Business Processes // Proc. of the First International. Conf. "The Practical Application of Intelligent Agents and Multi-Agent Technology", London, UK. 1996. P. 345—376.
6. Sandholm T. Negotiation among self-interested computationally limited agents. Ph. D. Thesis // University of Massachusetts Amherst, 1996. URL: <http://www.cs.cmu.edu/~sandholm/dissertation.ps>. (дата обращения: 30.01.2017).
7. Smith R., and Davis R. Framework for Cooperation in Distributed Problem Solving // International Journal IEEE Transactions on Systems, Man and Cybernetics. 1981. Vol. 11. P. 61—70.
8. Contract Net Protocol. URL: <http://www.fipa.org/specs/fipa00030/SC00030H.html>. (дата обращения: 30.01.2017).
9. Бухвалов О. Л., Карсаев О. В., Самойлов В. В. Использование многоагентного подхода в MES-системах // Известия Кабардино-Балкарского научного центра РАН. 2001. № 1 (39). С. 133—136.
10. JXTA. URL: <https://en.wikipedia.org/wiki/JXTA> (дата обращения: 16.11.2016).
11. UPnP Forum. URL: <http://upnp.org/specs/> (дата обращения: 30.01.2017).
12. Nomadic Agent Working Group. URL: <http://www.fipa.org/subgroups/P2PNA-WG-docs/P2PNA-Spec-Draft0.12.doc>. (дата обращения: 30.01.2017).
13. Городецкий В. И., Карсаев О. В., Самойлов В. В., Серебряков С. В. Инструментальные средства для открытых сетей агентов // Известия РАН. "Теория и Системы Управления" 2008. № 3. С. 106—124.
14. Vollset E., Ingham D., Ezhilchelvan P. JMS on Mobile Ad-hoc Networks. URL: http://pdf.aminer.org/000/382/876/jms_on_mobile_ad_hoc_networks.pdf. (дата обращения: 30.01.2017).
15. OASIS standard. Web Services Business Process Execution Language Version 2.0, 11 April 2007. URL: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>. (дата обращения: 30.01.2017).
16. Smith G. Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver // International Journal IEEE Transactions on Computers. 1980. V. 29. N 12. P. 1104—1113.
17. Serugendo D.-M., Gleizes G., and Karageorgos M.-P. Self-organization in multi-agent systems // Rapport de recherche IRIT/2005-18-R, IRIT, Universite Paul Sabatier, Toulouse. URL: http://www.irit.fr/TFGSO/DOCS/TFG2/TFGIISO_Long-Report.pdf. (дата обращения: 30.01.2017).
18. Городецкий В. И. Самоорганизация и многоагентные системы. Модели многоагентной самоорганизации // Известия РАН. Теория и системы управления. 2012. № 2. С. 92—120.
19. Городецкий В. И. Самоорганизация и многоагентные системы. II. Приложения и технология разработки // Известия РАН. Теория и системы управления. 2012. № 3. С. 55—75.
20. Jelasity M. and Babaoglu O. T-Man: Gossip-Based Overlay Topology Management // Series Lecture Notes in Artificial Intelligence. v. 3910 / Eds. Brueckner S. A. et al. Springer. 2006. P. 1—15.
21. Clarke I., Sandberg O., Wiley B., Andhong T. W. Freenet: A distributed anonymous information storage and retrieval system // Proc. International Workshop on Designing Privacy Enhancing Technologies. Springer, 2000. P. 46—66.
22. Noy N. F. Semantic Integration: A Survey Of Ontology-Based Approaches // Sigmod Record, Special Issue on Semantic Integration. 2004. URL: <http://www.ontologymatching.org>. (дата обращения: 30.01.2017).
23. Roser S. Ontology-based Model Transformation // ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, 2005.

Self-Organized B2B Production Networks. Part 2. Architecture and Algorithmic Support

V. I. Gorodetsky, gor@iias.spb.su, **O. L. Bukhvalov**, psychoveter@gmail.com,
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences,
St. Petersburg, 199178, Russian Federation

Corresponding author: **Gorodetsky Vladimir I.**, D. Sc.,
Chief Scientists of Intelligent Systems Laboratory, St. Petersburg Institute for Informatics
and Automation of the Russian Academy of Sciences, St. Petersburg, 199178, Russian Federation
e-mail: gor@iias.spb.su

Accepted on August 14, 2017

Network-centric business model implemented as open self-organizing B2B production network is a relatively novel paradigm of modern digital economy. It has become the subject of the active research and developments during the last times. In particular, this fact is proved by several dozen of the projects on this topic funded by the European Commission Programs during last 15 years. The developed countries consider this novel paradigm of production business organization as a solid trend and a new very promising future model for distributed business, virtual organizations and digital economy as a whole. The paper outlines and analyzes various aspects of this problem and their algorithmic support. In its Part II, the paper formulates the basic production management tasks to be solved in B2B production networks, those include distributed planning and distributed coordinated scheduling of resources of B2B production network nodes (particular companies, production plants), which resources are involved concurrently in multiple common production processes. The paper also analyses the basic (core and reusable) set of tasks and services of the infrastructure aimed to realize the self-organizing style of the B2B production network performance, to support of node interactions and communications, and to provide for the network with the openness. It outlines the basic algorithms solving the aforementioned tasks and refers to the previously developed software prototypes implementing the particular most complicated algorithms supporting B2B performance in stand-alone mode that justify the feasibility of the B2B paradigm of production network at current time.

Keywords: B2B networks, multi-agent architecture, self-organization, distributed coordination, software and communication infrastructure, planning, scheduling

Acknowledgments: The research is being performed according to the SPIIRAS Governmental Project № 0073-2015-0003 and the Project # 214 of the Program #1.5P of the Russian Academy of Sciences.

For citation:

Gorodetsky V. I., Bukhvalov O. L. Self-Organized B2B Production Networks. Part 2. Architecture and Algorithmic Support, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18, no. 12, pp. 829—839.

DOI: 10.17587/mau.18.829-839

References

1. **Bukhvalov O., Gorodetsky V., Karsaev O., Koudriyatshev G., Samoylov V.** *Proizvodstvennaya logistika: Strategicheskoe planirovanie, prognozirovaniye i upravleniye konfliktami* (Production Logistics: Strategic Planning and Conflict Management), *Transactions of South Federal University*, 2012, no. 3, pp. 209—218 (in Russian).
2. **Bukhvalov O., Gorodetsky V., Karsaev O., Koudriyatshev G., Samoylov V.** Privacy-Preserved Distributed Coordination of Production Scheduling in B2B Networks: A Multi-agent Approach, *Proc. 7th IFAC Conference on Manufacturing Modeling, Management, and Control*, 2013, vol. 7, part 1, pp. 2122—2127.
3. **Chavez A., Maes P.** Kasbah: An Agent Marketplace for Buying and Selling Goods, *Proc. First Intern. Conf. "The Practical Application of Intelligent Agents and Multi-Agent Technology"*, London, UK, 1996, pp. 75—90.
4. **Fisher K., Mueller J. P., Heimig I., Scheer A.-W.** Intelligent Agents in Virtual Enterprises, *Proc. First International. Conf. "The Practical Application of Intelligent Agents and Multi-Agent Technology"*, London, UK, 1996, pp. 205—224.
5. **Jennings N., Paratin P., Jonson M.** Using Intelligent Agents to Manage Business Processes, *Proc. of the First International. Conf. "The Practical Application of Intelligent Agents and Multi-Agent Technology"*, London, UK, 1996, pp. 345—376.
6. **Sandholm T.** Negotiation among self-interested computationally limited agents. Ph. D. Thesis, University of Massachusetts Amherst, 1996, available at: <http://portal.acm.org/citation.cfm?id=924453>, <http://www.cs.cmu.edu/~sandholm/dissertation.ps> (date of access 30.01.2017).
7. **Smith R., and Davis R.** Framework for Cooperation in Distributed Problem Solving, *International Journal IEEE Transactions on Systems, Man and Cybernetics*, 1981, vol. 11, pp. 61—70.
8. **Contract Net Protocol**, available at: <http://www.fipa.org/specs/fipa00030/SC00030H.html> (date of access 30.01.2017).
9. **Bukhvalov O., Karsaev O., Samoylov V.** *Ispol'zovaniye mno-goagentnogo podkhoda v MES-systemakh* (Multi-agent Approach to MES-Systems), *Transactions of Kabardino-Balkarsky Scientific Center of the Russian Academy of Sciences*, 2011, no. 1 (39), pp. 133—136 (in Russian).
10. **JXTA**, available at: <https://en.wikipedia.org/wiki/JXTA> (date of access 30.01.2017).
11. **UPnP Forum**, available at: <http://upnp.org/specs/> (date of access 30.01.2017).
12. **Nomadic Agent Working Group**, available at: <http://www.fipa.org/subgroups/P2PNA-WG-docs/P2PNA-Spec-Draft0.12.doc> (date of access 30.01.2017).
13. **Gorodetsky V., Karsaev O., Samoylov V., Serebrykov S.** Development tools for open agent networks, *Journal of Computer and Systems Sciences International*, 2008, no. 3, vol. 47, pp. 429—446 (in Russian).
14. **Vollset E., Ingham D., Ezhilchelvan P.** JMS on Mobile Ad-hoc Networks, available at: http://pdf.aminer.org/000/382/876/jms_on_mobile_ad_hoc_networks.pdf. (date of access: 30.01.2017).
15. **OASIS standard.** Web Services Business Process Execution Language Version 2.0, 11 April 2007, available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf> (date of access: 30.01.2017).
16. **Smith G.** Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver, *International Journal IEEE Transactions on Computers*, 1980, vol. 29, no. 12, pp. 1104—1113.
17. **Serugendo D.-M., Gleizes G., and Karageorgos M.-P.** Self-organization in multi-agent systems, *Rapport de recherche IRI/2005-18-R, IRI/IT*, Université Paul Sabatier, Toulouse, available at: http://www.irit.fr/TFGSO/DOCS/TFG2/TFGIISO_LongReport.pdf. (date of access: 30.01.2017).
18. **Gorodetskii V. I.** Self-Organization and Multi-agent Systems: I. Models of Multi-agent Self-Organization, *Journal of Computer and Systems Sciences International*, 2012, vol. 51, no. 2, pp. 256—281 (in Russian).
19. **Gorodetskii V. I.** Self-Organization and Multi-agent Systems: II. Applications and the Development Technology, *Journal of Computer and Systems Sciences International*, 2012, vol. 51, no. 3, pp. 391—409 (in Russian).
20. **Jelasy M., Babaoglu O.** T-Man: Gossip-Based Overlay Topology Management, *Series Lecture Notes in Artificial Intelligence*. v. 3910 / Eds. Brueckner S. A. et al., Springer, 2006, pp. 1—15.
21. **Clarke I., Sandberg O., Wiley B., Andhong T. W.** Freenet: A distributed anonymous information storage and retrieval system, *Proc. International Workshop on Designing Privacy Enhancing Technologies*, Springer, 2000, pp. 46—66.
22. **Noy N. F.** Semantic Integration: A Survey Of Ontology-Based Approaches, *Sigmod Record, Special Issue on Semantic Integration*, 2004, available at: <http://www.ontologymatching.org>. (date of access: 30.01.2017).
23. **Roser S.** Ontology-based Model Transformation, *ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, 2005.