

ПРИНЦИПЫ ДИНАМИЧЕСКОГО КОНФИГУРИРОВАНИЯ ГИБКИХ СЕТЕЙ ПОСТАВОК НА ОСНОВЕ СЕМАНТИЧЕСКОЙ КОМПОЗИЦИИ СЕРВИСОВ

СМИРНОВ А.В., ШЕРЕМЕТОВ Л.Б.

УДК 62-519

Смирнов А.В., Шереметов Л.Б. Принципы динамического конфигурирования гибких сетей поставок на основе семантической композиции сервисов.

Аннотация. В сложных бизнес-сетях поиск партнера может занимать довольно много времени. В сетях поставок, работающих на заказ, эта задача должна решаться в сжатые сроки и в условиях неопределенности, как по поставщикам, так и по заказам. Технологии семантических сервис-ориентированных архитектур могут обеспечить поддержку решения такого рода задач, позволяя строить самоорганизующиеся гибкие сети поставок. В статье описан подход к автоматизации динамического поиска и выбора участников сети, основанный на описании профилей их компетенций как сервисов. Предложенный подход использует метод поиска сервисов с неполной информацией и механизмы расширения запросов. Подход проиллюстрирован на примере из автомобильной промышленности.

Ключевые слова: гибкие сети поставок, конфигурация, семантические сервис-ориентированные архитектуры, поиск и композиция сервисов.

Smirnov A.V., Sheremetov L.B. Principles of Dynamic Configuration of Flexible Supply Networks Based on Semantic Service Composition.

Abstract. In a complex business network finding a supplier can be a very time consuming task. In advanced supply networks like Build-to-Order supply chains, this task should be carried out under time constraints and under uncertainties both in suppliers and in the orders. The technology of semantic service oriented architectures is aimed to support such kind of tasks, enabling construction of self-organizing flexible supply networks. A novel approach to dynamic network members' discovery and selection based on competence profiles included in provided service descriptions is proposed. The approach is grounded in a method for service discovery with incomplete information using query expansion techniques. The approach is illustrated by example from automotive industry.

Keywords: supply network, configuration, semantic SOA, service discovery and composition.

1. Введение. В современной экономике производственные системы мигрируют от традиционных предприятий к сетевым группам технологически связанных компаний и, более того, от интегрированных предприятий с фиксированной структурой к сетям поставок с переменной структурой, основанным на распределенном взаимодействии участников. Такие группы предприятий уже управляются не централизованно компаниями-лидерами, а через виртуальные пространства принятия решений. В этих условиях такие свойства, как гибкость и адаптивность играют ключевую роль в обеспечении конкурентоспособности предприятий [6, 12].

Следствием этого процесса является появление нового типа производственных систем, известных как гибкие сети поставок (ГСП), способные постоянно менять свою структуру и отношения между партнерами, адаптируясь к условиям внешней среды и изменениям в самой системе [5, 10, 15, 16]. С целью обеспечения такой гибкости развиваются новые стратегии построения сетей поставок, как например, сети, работающие на заказ (*build-to-order*), когда сначала в систему поступают заказы и по ним уже производится производство и сборка изделий [13, 14]. По данным отчета AMR Research «Руководство по управлению спросом», передовые компании, ориентирующиеся на заказы, держат на 15% меньше запасов, выполняют на 17% больше заказов и имеют цикл обращения денежных средств на 35% короче, чем у других компаний [9]. Например в автомобильной промышленности наибольшим преимуществом обладают те компании, которые могут выпускать автомобили, собранные по требованию заказчика в минимальные сроки. При этом в задачи системы управления конфигурацией входит динамическое построение каналов производства и поставки изделия для каждой заявки.

Эффективное управление компетенцией является одним из основных требований для эффективной конфигурации ГСП [18]. Анализ профиля предприятия на этапе конфигурирования ГСП требует динамической обработки больших объемов информации и знаний, размещенных в распределенных гетерогенных источниках. Как правило, получение необходимого результата связано с выборкой контекста из нескольких источников, что в условиях различия форматов, протоколов взаимодействия и стандартов описания приводит к необходимости решения задач семантической совместимости.

На техническом уровне совместимость поддерживается сервисориентированными архитектурами (COA), а на семантическом уровне – онтологиями. Подход, описываемой в данной статье, заключается в интеграции этих уровней в семантическую COA. И если достоинства данной архитектуры в задачах интеграции приложений на уровне предприятия уже рассматривались в литературе [7, 11], то возможности их использования для автоматизации процессов управления ГСП еще только предстоит исследовать. При этом могут рассматриваться различные задачи в рамках самоорганизации ГСП, такие как самоконтекстуализация, оптимизация, конфигурация, сглаживание противоречий, защита и т. д. [2, 4].

Для того чтобы автоматизация конфигурирования ГСП стала возможной, необходимо сформулировать и решить три ключевые задачи:

1) обеспечить синтаксическую и семантическую совместимость сервисов в рамках сети поставок,

2) автоматизировать процессы поиска и выбора наиболее предпочтительного сервиса, основанные на анализе синтаксических и семантических описаний,

3) обеспечить адекватную композиция сложных сервисов, определяющую конфигурацию сети.

В статье разработана концептуальная модель информационной среды ГСП, основанной на семантической СОА, интегрирующей технологии управления онтологиями и знаниями, профилирования, web-сервисов и агентов. Концептуальная модель использована для решения задачи динамического конфигурирования ГСП на основе семантической интеграции сервисов, предоставляемых участниками сети. Используемые технологии заимствованы из семантического web, целью которого является обеспечение интеллектуального анализа web-контента: данных и знаний. В основе разработанного подхода лежат методы поиска web-сервисов с использованием техники расширения запросов на онтологии предметной области, широко применяемой в информационно-поисковых системах [17].

2. Постановка задачи исследования. В статье рассматривается задача динамического конфигурирования ГПС под заказ на основе профилей компетенции участников сети. Определим компетенцию как способность компании выполнять бизнес-процессы поддерживаемые необходимыми ресурсами, практиками и действиями с целью производства продукции и предоставления сервисов. Компетенция компании описывается в ее профиле [20]. Типовой профиль участника сети, представленный на рис. 1, содержит следующую информацию:

- данные о компании;
- данные о филиале;
- историю заказов, которая хранит информацию по всем заказам за определенный период;
- предпочтения относительно продуктов и сервисов.

При этом часть информации о компетенции может быть непосредственно использована на этапе конфигурирования сети:

- возможности, которые описывают продукцию, поставляемую компанией, и предоставляемые сервисы;
- объемы производства и сроки поставок, которые используются для определения соответствия потенциального участника ограничениям по заказу;

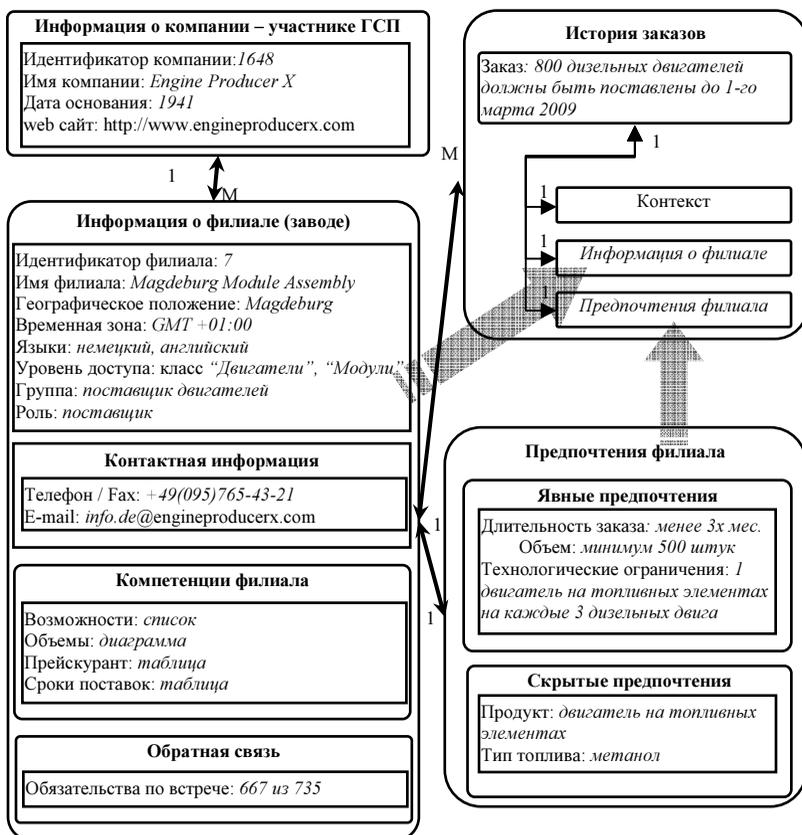


Рис. 1. Концептуальная модель профиля компетенции участника сети.

- прейскурант цен, который может быть использован для снижения стоимости производства в сети;
- географическое положение участника сети, которое используется для определения времени доставки и доступности предприятия в данный период.

Дополнительно могут быть учтены предпочтения компании:

- время заказа (краткосрочные и долгосрочные заказы);
- объем заказа (единичное или массовое производство);
- технологические ограничения (например, совместное производство различных наименований продукта).

Ставится задача автоматизировать конфигурирование сети на основе анализа профилей предприятий в рамках контекста запроса и построения такой конфигурации ГСП, которая бы удовлетворяла заданным ограничениям.

3. Базовые технологии построения информационной среды ГСП. Решение сформулированной задачи строится с использованием технологий семантической СОА (технологий агентов, web-сервисов и управления онтологиями). Каждый участник сети выступает как независимый источник знаний, доступ к которым обеспечивается как посредством web-сервисов, так и сервисов интеллектуальных агентов. Анализ запроса производится с использованием онтологии сети [1]. Обоснуем выбор перечисленных технологий.

При моделировании ГСП естественным образом возникает виртуальное представление участников сети с помощью программных агентов, выполняющих функции предварительного семантического анализа информации и ведения переговоров между партнерами [19]. Агенты используются как высокоуровневая абстракция для формализации и структурирования предметной области и как мощное программное средство для разработки и реализации сложных информационных систем. Разработанные специально для коллективного решения задач, сервисы агентов могут дополнять существующие системы планирования ресурсов предприятия (ERP), обеспечивая взаимодействие на более высоком уровне, чем транзакциональные данные, тем самым, придавая гибкость системам принятия решений в ГСП.

Задача совместимости и взаимодействия между информационными системами предприятий сети традиционно причисляется к одной из ключевых для построения интегрированных самоорганизующихся ГПС из-за наличия большого числа различных стандартов, с одной стороны, и высокой стоимости разработки различного рода интеграторов. Web-сервисы, описываемые на языке WSDL, технически решают проблему совместимости, однако не обладают свойствами семантического описания. Семантическая разметка предоставляет информацию о том, что и как делает данный сервис и необходима для автоматизации процессов поиска, вызова, композиции и мониторинга сервисов. Для добавления семантики существуют несколько альтернатив описания, наиболее распространенной из которых является основанный на XML язык OWL-S (*Web Ontology Language for Services*) [22]. Онтология OWL-S включает три основных части: профиль, процесс и обоснование (*profile, process, grounding*). Профиль описывает функциональные свойства сервиса, которые необходимы для его автоматического обна-

ружения, такие, например, как предложение сервиса, его входы и выходы, предварительные условия и дополнительные действия. На основе профиля, который представляет информацию о функциональных возможностях и функциональных атрибутах сервиса, создаются описания и запросы сервиса. Процесс дает детальное описание функционирования сервиса, тогда как обоснование описывает формы взаимодействия с сервисом посредством механизма сообщений. OWL-S выбран в данной статье для описания семантики сервисов, потому что обладает рядом преимуществ по сравнению с WSDL-S и WSMO:

- в практических целях обоснование привязано к стандарту WSDL и открыто для использования других языков;

- в профиле сервиса можно наряду с функциональными определить и нефункциональные свойства, такие как профиль провайдера, качество, цена, географическое положение и др.;

- также в профиле обычно определяется категория сервиса (информационный, производственный и т. д.), что является необходимым для композиции;

- модель процесса может быть использована для хранения информации о конфигурации сложного сервиса, что в рамках рассматриваемой задачи позволяет хранить конфигурацию сети и отношения между ее участниками.

Исследования по семантическому поиску сервисов стремительно развиваются в последние годы [21]. Методы поиска сервисов, основанные на сравнении запроса пользователя с описанием web-сервиса, особенно важны тогда, когда запрашиваемый сервис не найден или наоборот существует несколько реализаций сервиса. В обоих случаях необходимы методы определения сходства сервисов и их ранжирования на основе меры сходства. Большинство существующих подходов ограничивается только полным сходством функциональных свойств сервиса: входов, выходов, предусловий и эффекта (IOPE). Для его определения, как правило, используются методы совпадения ключевых слов и логического вывода [3, 21]. Подход, описанный в данной статье, ориентирован на поиск сервисов с неполным и нечетким совпадением. Для сравнения используются как функциональные, так и нефункциональные свойства, которые описывают профиль участников ГСП.

Использование сервисов web и агентов потребовало расширения языка семантического описания web-сервисов (OWL-S) на область сервисов агентов (OWL-AS), совместимое с требованиями стандарта FIPA [8]. Основная модификация OWL-S состоит в добавлении к обоснованию нового типа доступа к сервису, в котором возможно

регистрировать информацию о версии языка взаимодействия агентов FIPA-ACL, определять протоколы взаимодействия, используемые для вычисления сервиса, язык описания содержания сообщений и дополнительную информацию по локализации агента, предоставляющего сервис. Детали семантического описания сервисов агентов выходят за рамки данной статьи.

Для реализации подхода целесообразно использование онтологий, представляющих собой детальное описание предметной области, которое используется для формального и декларативного определения ее концептуализации. Алгоритмы нахождения соответствия между запросом и сервисом, которые используют онтологическое представление знаний, позволяют автоматизировать нахождение семантического подобия между запросом и описанием сервиса, несмотря на синтаксические различия между ними. Более того, как показано в дальнейшем, наличие явного представления знаний о предметной области, к которой относится сервис, допускает переформулировку запросов контекстно-зависимым способом, а впоследствии и переговоры о возможностях этого сервиса.

4. Концептуальная модель информационной среды конфигурирования ГСП. Процесс конфигурирования ГСП под заказ начинается с того, что новая заявка поступает в информационную систему обработки заказов, которая направляет его всем предприятиям, способным его выполнить. Зачастую речь идет о достаточно большом числе иерархически организованных предприятий, расположенных в различных географических регионах. При этом каждый участник сети выбирает своих прямых поставщиков, т. е. поставщиков «первого уровня». Так, в соответствии с рис. 2, участник D выбирает поставщиков из A, B и C, а участник G может выбирать поставщиков из D и E. Для выбора потенциальных поставщиков используются семантические описания их профилей и онтология сети, определяющая отношения между продуктами и предприятиями (например, отношение «производится на») [20].

С целью автоматизации рассмотренной общей схемы разработана архитектура сервис-портала ГСП (рис. 2). Разработка сервис-портала является развитием комплекса алгоритмических и программных средств динамического конфигурирования сетей поставок [19]. Все участники сети регистрируют свои профили как сервисы в реестре сервисов и службе каталогов агентной платформы соответственно. Эти реестры описаний используются вместо традиционных UDDI, обеспечивая тем самым возможность применения разработанных в статье

методов и алгоритмов поиска, сравнения и композиции. Их использование позволяет информационным системам участников сети обмениваться прямыми запросами (без вмешательства пользователя) через сервис-портал. Агенты в рассматриваемом подходе также используются как промежуточная технология, обеспечивающая, с одной стороны, механизмы поиска, с другой – взаимодействие web-сервисов с сервисами агентов.

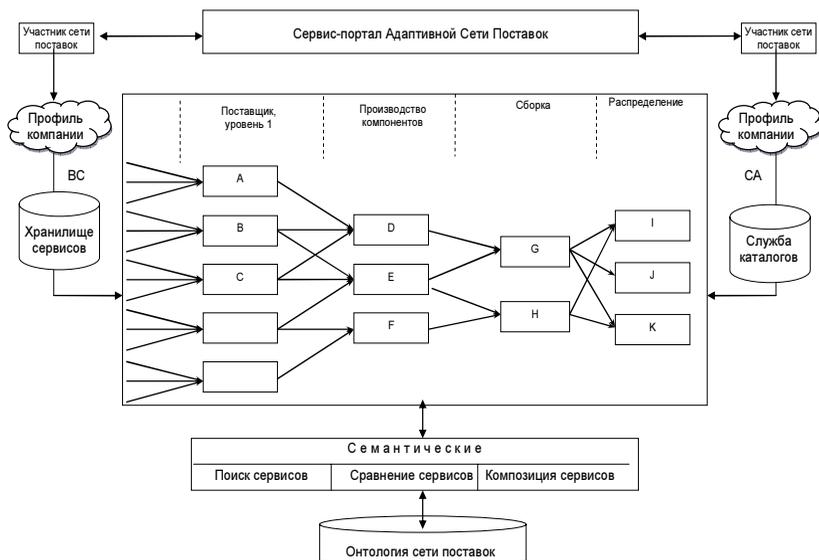


Рис. 2. Концептуальная модель информационной среды самоорганизующейся ГСП основанной на семантической COA.

Предложенная модель поиска партнеров сети работает следующим образом (рис. 3). Заявка на создание нового канала под заказ генерируется для поиска сервисов, предоставляемых участниками ГСП. Предложенный подход использует агента-брокера, в функции которого входит генерация и поиск сходных описаний в различных источниках и их последующее ранжирование. Запрос посылается агенту-брокеру с соответствующим описанием на OWL-S. Агент сравнивает запрос с доступными описаниями, зарегистрированными как web или агентные сервисы, с целью определить тип и степень сходства для последующего упорядочения и выбора сервисов с максимальной степе-

ную подобия. Сначала ищется такой сервис, который бы удовлетворял условиям равенства функциональных свойств (полное соответствие), к которым относятся входы, выходы и классификация. Если полного соответствия не найдено, т. е. требуемый сервис отсутствует, недоступен в настоящее время или существует неопределенность в его описании, система переходит к генерации семантически схожих описаний для поиска наиболее сходного с запрашиваемым. Семантически схожие описания посылаются доступным реестрам описаний или провайдерам. На обратном пути агент получает описания найденных сервисов для фильтрации и упорядочения по степени семантического сходства. Когда процесс поиска заканчивается, в тех случаях, когда простой сервис не найден и необходима генерация сложного сервиса, выполняется композиция сервисов, в результате которой формируется окончательная конфигурация канала сети поставок.

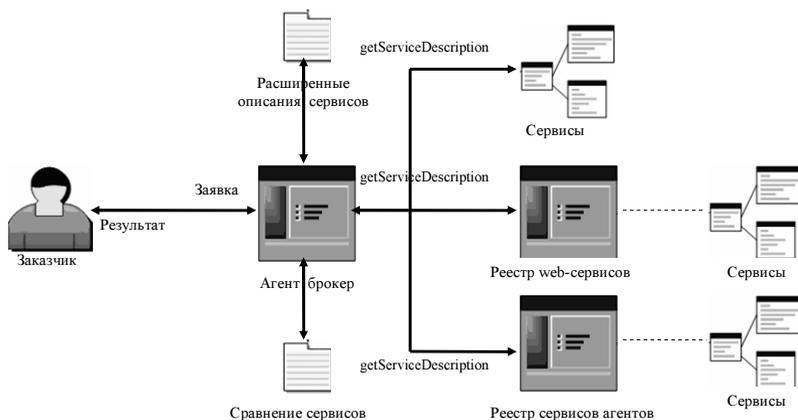


Рис. 3. Архитектура поиска сервисов.

Семантическая совместимость сервисных интерфейсов обеспечивается семантической моделью COA [17]. Эта модель включает в себя определение подобия между запрашиваемыми и предоставляемыми сервисами, ранжирование найденных сервисов на основе нечетких степеней сходства, выбор участников сети с наиболее схожими профилями, удовлетворяющими условиям запроса и, наконец, композицию сервисов для динамического конфигурирования сети под заказ.

5. Семантическая модель описания сервисов. Формальная модель описания сервисов разработана с использованием дескриптивной

логики (ДЛ). ГСП состоит из множества участников, которые являются потребителями и провайдерами сервисов. С целью конфигурирования сети или производственного канала на заказ участники генерируют запросы s_C с целью нахождения сервиса s_Φ , предоставляемого провайдером. Рассмотрим три типа участников:

1) $A = \{a_1, \dots, a_{|A|}\}$ – множество программных агентов, которые используют либо предоставляют сервисы;

2) $U = \{u_1, \dots, u_{|U|}\}$ – все агенты (пользователи) систем, которые используют сервисы;

3) $\Gamma = \{\gamma_1, \dots, \gamma_{|\Gamma|}\}$ – множество участников, которые предоставляют web-сервисы.

Таким образом, участники сети образуют два пересекающихся множества: 1) клиенты, которые определяются как $C = U \cup A$ и используют сервисы; 2) провайдеры $\Phi = A \cup \Gamma$, которые предоставляют сервисы.

Пусть $WS = \{ws_1, \dots, ws_{|WS|}\}$ – множество всех web-сервисов. Эти сервисы предоставляются серверами web Γ , где каждый $ws_i \in WS$ принадлежит множеству унитарных классификаций ST_i . Web-сервис представлен кортежем из четырех элементов $ws_i = (I_i, O_i, P_i, E_i)$, в котором I_i представляет собой входы сервиса, O_i – выходы сервиса, P_i – предварительные условия или состояние σ окружающей среды¹ для того, чтобы сервис ws_i функционировал в ожидаемой манере и, наконец, E_i – эффекты либо состояние τ окружающей среды, в которое переходит среда из состояния σ в результате выполнения сервиса. Если рассматривать I как множество всех входов сервисов, O – множество всех выходов сервисов, P – множество всех предварительных условий сервисов и E – множество

¹ Говоря об окружающей среде, имеем в виду ограниченное представление, которое имеется о мире, окружающей обстановке или некоей области, описанной посредством онтологий понятий и сервисов.

всех эффектов сервисов, тогда соответственно можно сказать, что $I_i \subseteq I, O_i \subseteq O, P_i \subseteq P, E_i \subseteq E$.

Кроме вышеуказанных сервисов, мы располагаем $AS = \{as_1, \dots, as_{|AS|}\}$ – множеством всех сервисов программных агентов, в которых подобно web-сервисам каждый $as_j \in AS$ принадлежит унитарному множеству ST_j . Сервис агента описан кортежем из четырех элементов $as_j = (I_j, O_j, P_j, E_j)$, где I_j – входы сервиса, O_j – его выходы, P_j – предварительные условия, на которых ожидается правильное выполнение сервиса и E_j – эффекты, которые генерирует сервис as_j в результате исполнения. Соответственно, $I_j \subseteq I, O_j \subseteq O, P_j \subseteq P, E_j \subseteq E$. Обобщая вышесказанное, получим множество всех элементарных сервисов $S, S = WS \cup AS$.

Целью конфигурирования сети поставок является сложный сервис, определенный следующим образом.

Сложный сервис. Сложный сервис cs состоит из множества сервисов, $cs = S''$, которые будучи сконфигурированы неким образом (параллельно, последовательно или смешанно), обеспечивают некое множество выходов O'' на основе задания входов I'' , если выполняются все предусловия P'' всех сервисов, включенных в сложный сервис $cs = \{s_1, \dots, s_{|cs|}\}$, где $S'' \subseteq S, O'' \subseteq O, I'' \subseteq I$ и $P'' \subseteq P$. Каждый сложный сервис принадлежит множеству неунитарных классификаций ST'' . Если сервисы, составляющие cs со входами $\{I_1, \dots, I_{|cs|}\}$ и выходами $\{O_1, \dots, O_{|cs|}\}$, вычисляются параллельно, тогда $I'' = \bigcup_1^{|cs|} I_k$ и $O'' = \bigcup_1^{|cs|} O_k$.

Запрос клиента s_C генерируется с целью нахождения простого сервиса s_Φ , зарегистрированного провайдером в реестре, или генерации сложного сервиса $s_{C\Phi}$, в случае, если простой сервис не найден. Запрос s_C описан кортежем из четырех элементов $s_C = (S', \Psi', z, f)$,

где $S' \subseteq S$, Ψ' – компоненты, которые необходимы для вычисления ресурса (на практике это экземпляры входов $I' \subseteq I$); $z \in Z$ – критерий выбора сервиса и f – натуральное число, $f \in \mathbb{N}$, $f \neq 0$, – максимальное число опций, которые клиент желает получить. Если $f=1$, должен быть выбран единственный сервис, наилучшим образом удовлетворяющий условиям $z \in Z$. Определим критерий выбора как картезианское произведение условий

$$z = \prod_{l \in \{1, \dots, n\}} Q_l.$$

Схожесть между сервисами определяется на основе сопоставления функциональных свойств, т. е. входов, выходов и классификации, к которой они принадлежат. В процессе сопоставления сравнивается число входов и выходов и семантическое подобие концептов, которые их определяют [17]. При этом определены три типа сравнения сервисов: равенство, неравенство и сходство, которое, в свою очередь, можно представить тремя способами (операционное, входов и/или выходов). Наряду с традиционным полным сходством определены типы сходства с избыточной и недостающей информацией, что позволяет находить те сервисы, которые соответствуют (возможно, в некоторой степени) желаемым параметрам запроса. При дальнейшем описании подхода без потери общности ограничимся рассмотрением только входов и выходов.

Сравнивая запрошенные и предложенные сервисы, определим следующие типы множеств:

$$\ddot{I} = I_C - I_\Phi, \ddot{O} = O_\Phi - O_C, \underline{I} = I_\Phi - I_C \text{ и } \underline{O} = O_C - O_\Phi,$$

представляющие избыточные и недостающие входы и выходы соответственно. Возможные типы сходства представлены в табл. 1.

Чтобы определить сходство между сервисами, необходимо определить сходство элементов, которые его описывают, другими словами входов, выходов и классификаций, к которым они принадлежат. Так как описания сервисов основаны на OWL-S, семантическое сходство этих элементов определяется с использованием понятий *эквивалентности* и *производной*, определенных в ДЛ. Считается, что все элементы множества равны (другими словами $I_C = I_\Phi$ и $O_C = O_\Phi$), если на них может быть определено одно из следующих двух отношений, *эквивалентности* (\equiv) и *включения* (\sqsubseteq). Каждое отношение определяет тип сходства в соответствии с ДЛ. Каждый вход или выход представляется как частный случай концепта OWL из конкретной онтологии.

Таблица 1. Типы сходства сервисов

Тип сходства	Описание
а) Полное сходство сервисов	$I_C = I_\Phi$ & $O_C = O_\Phi$ & $ST_C = ST_\Phi$ & $\dot{I} = \ddot{O} = \dot{I} = \ddot{O} = \emptyset$
б) Полное сходство входов и выходов (I&O)	$I_C = I_\Phi$ & $O_C = O_\Phi$ & $\dot{I} = \ddot{O} = \dot{I} = \ddot{O} = \emptyset$
в) Сходство с избыточными входами	$I_\Phi \subseteq I_C$ & $\dot{I} \neq \emptyset$ & $\ddot{O} = \emptyset$
г) Сходство с избыточными выходами	$O_C \subseteq O_\Phi$ & $\dot{I} = \emptyset$ & $\ddot{O} \neq \emptyset$
д) Сходство с избыточными I&O	$I_\Phi \subseteq I_C$ & $O_C \subseteq O_\Phi$ & $\dot{I} \neq \emptyset$ & $\ddot{O} \neq \emptyset$
е) Сходство с недостающими входами	$I_C \subseteq I_\Phi$ & $\dot{I} \neq \emptyset$ & $\ddot{O} = \emptyset$
ж) Сходство с недостающими выходами	$O_\Phi \subseteq O_C$ & $\dot{I} = \emptyset$ & $\ddot{O} \neq \emptyset$
з) Сходство с недостающими I&O	$I_C \subseteq I_\Phi$ & $O_\Phi \subseteq O_C$ & $\dot{I} \neq \emptyset$ & $\ddot{O} \neq \emptyset$

Сопоставление между этими концептами относится к одному из следующих типов: а) *соответствие эквивалентности* или б) *соответствие включения*.

Пусть S_C – запрашиваемый сервис с входами I_C и выходами O_C . Так как i_{C_n} – один из входов сервиса S_C , где $i_{C_n} \in I_C$ и $1 \leq n \leq |I_C|$, считается, что вход i_{Φ_n} соответствует запрашиваемому входу, если $i_{C_n} \equiv i_{\Phi_n}$ или $i_{\Phi_n} \supseteq i_{C_n}$. Соответствие выходов определено аналогично: $o_{C_m} \equiv o_{\Phi_m}$ и $o_{C_m} \supseteq o_{\Phi_m}$, где $o_{C_m} \in O_C$ и $1 \leq m \leq |O_C|$.

Сходство сервисов измеряется с помощью их параметров. Степень сходства определяется на основе вычисления дистанции между концептами онтологии, в которой параметры сервисного описания являются концептами. При этом возможность использования лексической единицы A вместо лексической единицы B определяется входимостью A в иерархию гипонимов B либо наоборот (является ли A частным случаем B или наоборот). Несмотря на то, что в последнее время были предложены различные алгоритмы измерения дистанции между двумя концептами онтологии, в данной работе использована наиболее распространенная метрика определения расстояния как кратчайшего пути на семантической сети, введенная Р. Рада (Roy Rada) и его

коллегами в 1989 г. Алгоритм нахождения соответствия между понятиями запроса пользователя и описания сервиса, определенными на различных онтологиях приведен в [23].

Расстояние между входами и выходами измеряется следующим образом. Вход i_{c_m} имеет нулевое расстояние до входа i_{Φ_m} , если оба входа описаны одним и тем же концептом. С другой стороны, i_{c_m} имеет расстояние -1 до входа i_{Φ_m} , если i_{Φ_m} – предок концепта i_{c_m} . Выход o_{c_n} имеет нулевое расстояние до выхода o_{Φ_n} , если оба выхода описаны одним и тем же концептом. С другой стороны, o_{c_n} имеет расстояние равное 1 до выхода o_{Φ_n} , если этот выход – потомок концепта o_{c_n} . Таким же образом расстояния вычисляют для любого найденного уровня потомка или предка, применяя техники восхождения и нисхождения (*upcasting and downcasting*) соответственно для входов и выходов. Для того чтобы рассчитать расстояние между сервисными описаниями клиента и поставщика, рассчитываются соответствующие расстояния для каждого входа и выхода, затем абсолютные значения этих расстояний суммируются.

С целью определения семантического сходства между сервисами рассмотрим способ, основанный на теории нечетких множеств. При этом степени принадлежности каждой функции помогают определить сходство между двумя сервисами с целью выбора сервиса. Функция принадлежности для входов $\mu_{i_{c_m}}(i_{\Phi_m})$, которая задает степень сходства входа i_{Φ_m} найденного сервиса по отношению к входу запрошенного сервиса i_{c_m} , определена следующим образом:

$$\mu_{i_{c_m}}(i_{\Phi_m}) = \begin{cases} \frac{2}{(1 + e^{\text{paren_average}^{i-1} \cdot \text{updis} \cdot \text{distance}})^{-1}} & \text{if } i_{c_m} = i_{\Phi_m}, \\ 0 & \text{otherwise} \end{cases}$$

где *paren_average* – усредненная максимальная глубина онтологии; *i-updis distance* – это расстояние между входом описания клиента i_{c_m} по отношению ко входу найденного сервиса i_{Φ_m} , полученное путем использования техники восхождение.

Функция принадлежности μ_{I_C} определяет три степени сходства (низкую, среднюю и высокую) между множествами входов I_C и I_Φ следующим образом:

$$\mu_{I_C}(I_\Phi) = \{\mu_{I_{CL}}, \mu_{I_{CM}}, \mu_{I_{CH}}\},$$

где:

$$\mu_{I_{CL}}(I_\Phi) = \begin{cases} 1, & \text{если } \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) < \frac{|I_C|}{3} \\ \frac{\left(\frac{|I_C|}{3} \cdot 2\right) - \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right)}{\left(\frac{|I_C|}{3} \cdot 2\right) - \left(\frac{|I_C|}{3}\right)}, & \text{если } \frac{|I_C|}{3} \leq \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \leq \left(\frac{|I_C|}{3} \cdot 2\right) \\ 0, & \text{если } \left(\frac{|I_C|}{3} \cdot 2\right) < \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \end{cases}$$

$$\mu_{I_{CM}}(I_\Phi) = \begin{cases} \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) - \frac{|I_C|}{3}, & \text{если } \frac{|I_C|}{3} \leq \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \leq \left(\frac{|I_C|}{3} \cdot 2\right) \\ \frac{|I_C| - \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right)}{3}, & \text{если } \left(\frac{|I_C|}{3} \cdot 2\right) \leq \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) < |I_C| \\ 0 & \text{if } \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) < \frac{|I_C|}{3} \text{ или } |I_C| < \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \end{cases}$$

$$\mu_{I_{CH}}(I_\Phi) = \begin{cases} \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) - \left(\frac{|I_C|}{3} \cdot 2\right), & \text{если } \left(\frac{|I_C|}{3} \cdot 2\right) \leq \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \leq |I_C| \\ 1, & \text{если } |I_C| \leq \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) \\ 0, & \text{если } \left(\sum_1^{|\mathcal{I}_C|} \mu_{i_{C_m}}(i_{\Phi_m})\right) < \left(\frac{|I_C|}{3} \cdot 2\right) \end{cases}$$

В случае, когда не удастся обнаружить сервис с полным сходством описания с запрашиваемым клиентом, необходимо сгенерировать расширенные сервисные описания. Для этого разработаны два алгоритма расширения описаний:

– первый включает в себя генерацию всех возможных (в пределах предметной онтологии) описаний сходных сервисов, поиск таких сервисов и затем оценку, классификацию и выбор наиболее сходных сервисов. Этот метод называется исчерпывающей генерацией описаний.

– второй алгоритм включает генерацию запросов в зависимости от степени сходства последовательно, начиная с наиболее сходного запроса и заканчивая наименее сходным. При этом в случае нахождения сервисов, их можно выбрать непосредственно без дополнительной оценки и классификации. Этот метод называется пошаговой генерацией.

С деталями алгоритмов можно ознакомиться в [17].

6. Описание профилей сервисов. Наряду с функциональными свойствами (IOPE) профиль сервиса, заданный на OWL-S, может включать в себя нефункциональные свойства, которые позволяют семантически описать дополнительную информацию о компании: профиль ее компетенций, данные о владельце, контактную информацию, рейтинг, показатели качества, географическое положение и т. д. (рис. 1). Как показано в табл. 2, в профиле сервиса различают общие (*generalInfo.owl*) и специфические (*specificInfo.owl*) свойства. Общие свойства связаны с информационным профилем с помощью тэга *actor:webURL*, тогда как специфические – посредством тэга *company:info*. Важно отметить, что для общих свойств можно определить различные специфические профили. В свою очередь категория сервиса может быть определена в тэге *profileHierarchy*, при этом концепт *иерархия (Hierarchy)* в онтологии категорий сервисов открыт для расширения, что позволяет ввести новые типы сервисов, например информационные или производственные. В процессе конфигурирования нефункциональные свойства используются как дополнительный критерий выбора сервиса.

7. Динамическая композиция сервисов. Динамическая конфигурация ГСП может рассматриваться как расширение техник композиции сервисов. Так, необходимость в новом поставщике некоего сырья сходна с поиском нового сервиса для обеспечения недостающего входа. Композиция сервиса sc_A , где $1 \leq i \leq |sc_A|$, включает четыре этапа: 1) выбор сервиса, 2) генерирование расширенных сервисных описаний, 3) обнаружение сервиса и 4) оценку соответствия сервисных описаний. Ниже приведен алгоритм динамической композиции, основанный на семантическом расширении описаний сервисов:

Э т а п 1. Выбрать сервисы s_ϕ или sc_ϕ , которые имеют отношение к запросу клиента sc_i :

- имеют большее число выходов (недостающие выходы);
- имеют меньшее число входов (недостающие входы);

Таблица 2. Фрагмент спецификации профиля компетенции участника ГСП

GetSFCMXInformationProfile.owl	generalInfo.owl	specificInfo.owl
<pre> ... <actor:Actor rdf:ID=" Magdeburg -owner"> <actor:name> Magde- burg </actor:name> <actor:title> MagdeburgSupplyWS Creator and Owner </actor:title> <ac- tor:phone>49(095)765 43-21 </actor:phone> <ac- tor:fax>49(095)765- 43-21</actor:fax> <actor:email> info.de@engineproducer x.com </actor:email> <ac- tor:physicalAddress> Magdeburg </actor:physicalAddre ss> <actor:webURL> 192.168.1.254/ generalInfo.owl </actor:webURL> </actor:Actor>... </pre>	<pre> <!-- Company General information --> <pro- file:companyInformation > <company:Company rdf:ID="Magdeburg01"> <company:name> Mag- deburg Plant</company:name> <com- pany:foundation>03/23/ 1941</company:foundati on> <company:homePage> http://www.engineprodu cerx.com </company:homePage> <com- pany:SpecificInformatio n> <com- pany:info>192.168.1.254 / specificInfo.owl </company:info> </company:SpecificInfor mation> </company:Company> </profile:companyInfor mation> </pre>	<pre> <profile:specificInformation> <specificCompany:Facility rdf:ID="fac01"> <specificCompany:facilityName> Magdeburg_Module_Assembly </specificCompany:facilityName> <specificCompany:location> Mag- de- burg</specificCompany:location> <specificCompany:Preferences rdf:ID="prf01"> <companyPreferences:time> short- term </companyPreferences:time> <companyPrefer- ences:volume>serial production </companyPreferences:volume> <companyPrefer- ences:fuelType>methanol</compan yPreferences:volume> <companyPrefer- ences:fuelType>petrol</companyPr eferences:volume> <companyPrefer- ences:capabilityRestrictions > </ companyPrefer- ences:capabilityRestrictions> </specificCompany:Preferences> </specificCompany:Facility> </profile:specificInformation> </pre>

– имеют большее число выходов и меньшее число входов (недостающие I&O).

Добавить s_{Φ_i} или каждый элемент sc_{Φ_i} к sc_A .

Э т а п 2. Расширить запрос клиента $s_{C_{i+1}}$ таким образом, чтобы входы $I_{C_{i+1}}$ полностью соответствовали исходному запросу, $I_{C_{i+1}} = I_C$, а выходы – $O_{C_{i+1}}$:

– $O_{C_{i+1}} = O_i$, недостающие выходы сервиса s_{Φ_i} либо sc_{Φ_i} относительно sc_i ;

– $O_{C_{i+1}} = I_i$, недостающие входы сервиса s_{Φ_i} либо sc_{Φ_i} относительно sc_i .

Это означает, что каждый новый запрос будет искать сервис s_{Φ_i} или sc_{Φ_i} , при этом $s_{\Phi_{i+1}}$ или каждый элемент $sc_{\Phi_{i+1}}$ будут прибавляться к sc_A .

Э т а п 3. Применить один из алгоритмов обнаружения сервисов.

Э т а п 4. Оценить сходство: если существуют сервисы, соответствующие sc_A , и тип сходства между s_C и sc_A , который относится к:

– сходству с недостающими выходами, недостающими входами или недостающими I&O, тогда вернуться к этапу 1;

– полному сходству или сходству с избыточной информацией, тогда композиция закончена.

Алгоритм заканчивается тогда, когда достигнуто одно из следующих условий:

– не существует сервисов, предлагающих недостающие входы и выходы,

– число попыток или время истекли.

Нефункциональные свойства профиля сервисов (табл. 2) оцениваются в случае обнаружения нескольких поставщиков. На этом этапе задаются правила предпочтения, как, например (в SQL нотации): *Select (companyPreferences:fuelType = "methanol")*.

8. Иллюстрация подхода. Проиллюстрируем реализацию задачи выбора возможных поставщиков ГСП на примере, адаптированном из проекта «Интеллектуальная логистика для инновационных производства» (ILIPT, www.ilipt.org). Предположим, что сборочному производству поступил новый заказ на сборку 200 штук двигателей мощностью 150кВт на топливных элементах на основе метанола. Заказ описывается онтологией двигателей, фрагмент которой приведен на рис. 4.

На первом этапе определяется поставщик двигателей на топливных элементах. Предположим, что заказу соответствует запрос клиента со следующим описанием сервиса.

1. EngineSupplyWS(Fuel_cell_engine engine_input, Power power_input, Production_volume volume_input, Production_time time_input, Cost cost_output).

Предположим также, что по этому запросу найдены два описания сервисов для заводов, расположенных в Магдебурге и Марселе.

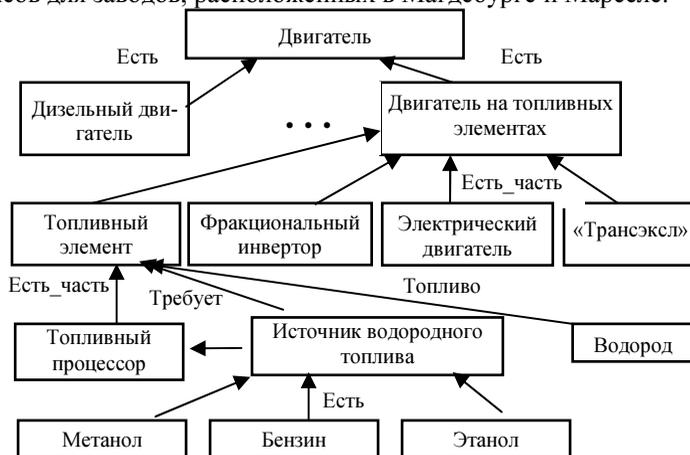


Рис. 4. Фрагмент онтологии двигателей.

2. *MagdeburgEngineSupplyWS(Fuel_cell_engine engine_input, Power power_input, Production_volume volume_input, Production_time time_input, Cost cost_output).*

3. *MarseilleEngineSupplyWS(Fuel_cell_engine engine_input, Power power_input, Production_volume volume_input, Production_time time_input, Cost cost_output).*

Эти описания имеют точное совпадение с запросом (1). В данном случае необходимо использовать предпочтения компаний, определенные в их профилях компетенции, чтобы выбрать требуемого поставщика. Как видно из табл. 2, завод в Магдебурге определяет типы топлива метанол и бензин <methanol, petrol>, тогда как для Марсельского заводы этот параметр определен как этанол и бензин <ethanol, petrol>. В результате сравнения будет выбран Магдебургский завод, так как он соответствует всем требованиям заказа. В результате вычисления сервиса определяется стоимость заказа, после чего необходимо идентифицировать поставщиков второго уровня. Описания производственного сервиса клиента и поставщика выглядят следующим образом.

4. *SupplyWS(Fuel_cell_engine engine_output).*

5. *SFCMXSupplyWS(Fuel_cell_cell_input, Fraction_inverter_inverter_input, Transaxle transaxle_input, Fuel_cell_engine engine_output).*

При сравнении описаний (4) и (5) очевидно, что описание (5) имеет три отсутствующих входа, которые соответствуют компонентам, требуемым для сборки. Еще один компонент, в соответствии с онтологией продукта – электрический двигатель – производится на самом предприятии. Процесс повторяется итеративно для поставщиков компонент двигателя (топливный процессор, фракционный инвертор, «Трансэксл») как показано ниже. Расширенные описания, соответствующие недостающим входам, генерируются автоматически на основе онтологии заказа (рис. 4), тем самым позволяя идентифицировать поставщиков недостающих компонентов (6) – (8).

6. *BremenSupplyWS(Fuel_processor processor_input, Fuel_cell cell_output,)*.

7. *PragueSupplyWS(Fraction_inverter inverter_output)*.

8. *PragueSupplyWS(Transaxle transaxle_output)*.

Для краткости показаны только значимые параметры. В описании (6) отсутствует один вход, что, в свою очередь, приводит к генерированию нового описания.

9. *PragueSupplyWS(Methanol_fuel_processor, processor_output)*.

Сравнивая (6) и (9), мы имеем семантически схожий выход (дистанция равна 1). На этом генерирование конфигурации канала, в который войдут заводы в Магдебурге, Бремене и Праге (рис. 5), может быть завершено.

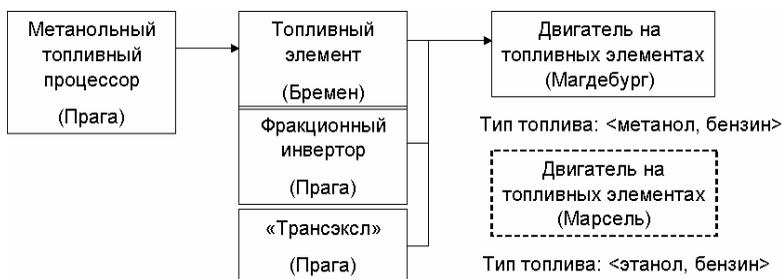


Рис. 5. Сгенерированная конфигурация ГСП.

9. Заключение. В статье рассмотрена важная проблема автоматизации процесса принятия решений при выборе партнеров и конфигурации ГСП на основе компетенции участников сети. Подход основан на представлении компетенции участников сети в терминах сервисов, которые они предоставляют, тем самым обеспечивается динамическое конфигурирование сети посредством семантической интеграции сер-

висов. Разработанные техники поиска позволяют работать с неполным сходством сервисов. Описание профиля компаний может представлять существенный интерес для планирования кооперации участников сети, обеспечивая автоматизацию процессов в бизнес-сетях. Возможными сферами приложения описанного подхода являются интеллектуальная поддержка принятия решений в системах управления компетенцией, а также построение самоорганизующихся ГСП.

Литература

1. *Angele J., Erdmann M., Wenke D.* Ontology-Based Knowledge Management in Automotive Engineering Scenarios // *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*. Vol. 7. Springer US, 2008. P. 245–264.
2. *Baumgarten M., Bicocchi N., Curran K., Mamei M. et al.* Towards Self-Organizing Knowledge Networks for Smart World Infrastructures. *Intern. Transactions on Systems Science and Applications*, 2006. Vol. 2, № 2. P. 123–133.
3. *Caceres C., Fernandez A., Ossowski S., Vasirani M.* Agent-Based Semantic Service Discovery for Healthcare: An Organizational Approach // *Intelligent Systems*. 2006. Vol. 21, № 6. P. 11–20.
4. *Chandran R., Hexmoor H.* Delegation Protocols Founded on Trust // *KIMAS'07: Modeling, Exploration, and Engineering (proc. of the 2007 Intern. Conf. on Integration of Knowledge Intensive Multi-Agent Systems)*. IEEE Press, 2007. P. 328–335.
5. *Choi Y. T., Dooley K. J., Rungtusanatham M.* Supply networks and complex adaptive systems: control versus emergence // *J. of Operations Management*. 2001. № 19. P. 351–366.
6. *Christopher M., Towill D.* An integrated model for the design of agile supply chains // *Intern. J. of Physical Distribution and Operations Management*. 2001. № 31. P. 235–244.
7. *Contreras M., Sheremetov L.* Industrial application integration using the unification approach to agent-enabled semantic SOA // *Robotics and Computer-Integrated Manufacturing*. 2008. Vol. 24, № 5. P. 680–695.
8. Foundation for Intelligent Physical Agents (FIPA) Documentation [Electronic resource] URL: <http://www.fipa.org> (access date: 24.06.2009).
9. *Fontanella J.* Awarding Excellence in B2B [Electronic resource] // *AMR Research: [site]*. [2007]. URL: <http://www.amrresearch.com> (access date: 24.06.2009).
10. *Garavelli A.* Flexibility configurations for the supply chain management // *Intern. J. of Production Economics*. 2003. № 85. P. 141–153.
11. *Gou J., Yang X., Dai W.* On Demand Integration of Dynamic Supply Chain Application Based on Semantic Service Oriented Architecture // *Research and Practical Issues of Enterprise Information Systems*. IFIP Book Series. 2008. Vol. 254. P. 589–598.
12. *Gunasekaran A., Lai K., Cheng T.* Responsive supply chain: a competitive strategy in a networked economy // *Omega*. 2008. № 36. P. 549–564.
13. *Gunasekaran A., Ngai N.* Build-to-order supply chain management: literature review and framework for development // *J. of Operations Management*. 2005. Vol. 23, № 5. P. 423–51.
14. *Kathawala Y., Wilgen A.* The evolution of build-to-order supply chain and its implications with selected case studies // *Intern. J. of Services and Operations Management*. 2005. Vol. 1, № 3. P. 268–282.
15. *Lummus R., Duclos L., Vokurka R.* Supply chain flexibility: Building a new model // *Global J. of Flexible Systems Management*. 2003. Vol. 4, № 4. P. 1–13.

16. *Pujawa N.* Assessing supply chain flexibility: A conceptual framework and case study // Intern. J. of Integrated Supply Management. 2004. Vol. 1, № 1. P. 79–97.
17. *Sanchez C., Sheremetov L.* A Model for Semantic Service Matching with Leftover and Missing Information // Proc. of the 8th Intern. Conf. on Hybrid Intelligent Systems, IEEE CS Press, California, US. 2008. P. 198–203.
18. *Sandkuh K., Smirnov A., Shilov N.* Configuration of Automotive Collaborative Engineering and Flexible Supply Networks // Expanding the Knowledge Economy. Issues, Applications, Case Studies. IOS Press, Amsterdam (NL). 2007. P. 929–936.
19. *Smirnov A., Sheremetov L., Chilov N., Sanchez-Sanchez C.* Agent-Based Technological Framework for Dynamic Configuration of a Cooperative Supply Chain // Multiagent-based supply chain management. Series on Studies in Computational Intelligence. 2006. Vol. 28. P. 217–246.
20. *Smirnov A., Shilov N., Kashevnik A.* Developing a knowledge management platform for automotive build-to-order production network // Human Systems Management. 2008. Vol. 27, № 31. P. 15–30.
21. *Sycara K., Paolucci M., Ankolekar A., Srinivasan N.* Automated Discovery, Interaction and Composition of Semantic Web Services // J. of Web Semantics. 2003. Vol. 1, № 1. P. 27–46.
22. OWL-S: Semantic Markup for Web Services [Electronic resource]. URL: <http://www.w3.org/Submission/OWL-S/> (access date: 24.06.2009).
23. *Розушина Ю.В., Гладун А.Я.* Онтологический подход к поиску веб-сервисов в распределенной среде Интернет // Информатика. 2006. № 4. С. 116–126.

Смирнов Александр Викторович — д-р техн. наук, проф.; заместитель директора по научной работе Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН), заведующий лабораторией интегрированных систем автоматизации. Область научных интересов: интеллектуальное управление конфигурациями виртуальных и сетевых организаций, логистика знаний, поддержка принятия решений. Число научных публикаций — 304. Адрес: smir@iiias.spb.su; СПИИРАН, 14-я линия В. О., д. 39, Санкт-Петербург, 199178, РФ; раб. тел. +7(812)328-2073, факс +7(812)328-4450.

Smirnov Alexander Victorovich — D.Sc., Prof.; a Deputy Director for Research and a Head of Computer Aided Integrated Systems Laboratory at St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), a full professor of St. Petersburg State Electrical Engineering University. Research interests: intelligent configuration management of virtual and network organizations, knowledge logistics, decision support. The number of publications — 304. Address: smir@iiias.spb.su; SPIIRAS, 39, 14th Line V. O., St. Petersburg, 199178, Russia; tel. +7(812)328-2073, fax: +7(812)328-4450.

Шереметов Леонид Борисович — канд. техн. наук, ст. науч. сотр. лаборатории интегрированных систем автоматизации Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Ассоциированный редактор журналов IEEE Transactions on Systems, Man and Cybernetics, Part C и "Computación y Sistemas". Член технических комитетов «Компьютерная поддержка коллективной работы в проектировании», IEEE Systems, Man and Cybernetics Society, и «Искусственный интеллект и экспертные системы», IASTED. Область научных интересов: многоагентные системы, гибридные интеллектуальные системы, поиск данных и семантические сервис-ориентированные архитектуры. Число научных публикаций — 250. Адрес: lsheremetov@mail.ru; СПИИРАН, 14-я линия В. О., д. 39, Санкт-Петербург, 199178, РФ; раб. тел. +7(812)328-8071, факс +7(812)328-4450.

Sheremetov Leonid Borisovich — PhD in Tech. Sciences, senior researcher, Computer Aided Integrated Systems Lab. at St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Associate Editor of the IEEE Transactions on Systems, Man and Cybernetics, Part C and "Computación y Sistemas". Member of the Technical Committees «Computer Supported Collaborative Work in Design» of the IEEE Systems, Man and Cybernetics Society and «Artificial Intelligence and expert Systems» of the IASTED. Research interests: multiagent systems, hybrid artificial intelligence, data mining and semantic SOA. The number of publications — 250. Address: lsheremetov@mail.ru; SPIIRAS, 39, 14-th Line V. O., St. Petersburg, 199178, Russia; tel. +7(812)328-8071, fax: +7(812)328-4450.

Поддержка исследований. В публикации представлены результаты исследований, поддержанные грантами следующих проектов: интегрированный проект FP6-IST-NMP 507592-2 "LIPIT", спонсируемый Европейской комиссией; грант #08-07-00264 РФФИ и проект № 213 Программы фундаментальных исследований Президиума РАН П2 "Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация".

Рекомендовано СПИИРАН, директор Юсупов Р.М., чл.-корр. РАН.
Статья поступила в редакцию 25.06.2009.