

Эта статья получена с сайта Артамонова Ивана Васильевича:

<http://artamonoviv.ru>

На сайте можно увидеть правильную (по ГОСТу) библиографическую ссылку на статью, там также доступны другие статьи этой же тематики, монографии и учебные пособия.  
Связаться с автором можно тоже через сайт.

Артамонов И.В.  
Аспирант

Байкальский государственный университет экономики и права,  
г. Иркутск

e-mail: [dark@darkis.ru](mailto:dark@darkis.ru)

## **Оркестровка и хореография: подходы к описанию композитных бизнес-процессов**

Выделяют два основных подхода к организации взаимодействующих бизнес-процессов: оркестровка и хореография. Оркестровка описывает поток взаимодействия между процессами организации, а хореография – последовательность условий такого взаимодействия. Статья приводит краткую характеристику этих подходов, анализирует их схожие и различные стороны и описывает их роль в сервис-ориентированной архитектуре.

Ключевые слова: оркестровка, хореография, веб-служба, бизнес-процесс, сервис-ориентированная архитектура, BPM, SOA.

Artamonov I.V.

Postgraduate

Baikal national university of economics and law

Irkutsk

e-mail: [dark@darkis.ru](mailto:dark@darkis.ru)

## **An orchestration and choreography: approaches to the description of composed business processes.**

There are two main approaches to the organization of interacting business processes: an orchestration and choreography. The orchestration describes the flow of an interaction between the processes of the organization, and the choreography - a sequence and conditions of the interaction. The article gives a brief description about these approaches, examines their similarities and differences, and describes their role in a service-oriented architecture.

Keywords: orchestration, choreography, web-service, business process, service-oriented architecture, BPM, SOA.

Идеология сервис-ориентированной архитектуры информационных систем и концепции управления бизнес-процессами предполагает, что отдельные программные службы обладают ограниченной функциональностью, а для решения сложных задач требуется использовать их различные комбинации. Выделяют два основных подхода к построению таких комбинаций: оркестровка и хореография.

Границы между оркестровкой и хореографией достаточно размыты. Хореографию иногда путают с оркестровкой и даже называют ею процессы оркестровки (к примеру, хореография бизнес-взаимодействия

стандарта ebXML BPSS не является таковой, а идеально вписывается в определение оркестровки) или целые программные системы (например, IBM WebSphere Process Server включает компонент, который называется «Business Process Choreographer» («Хореограф бизнес-процессов»), но этот компонент осуществляет моделирование бизнес-процессов с помощью языка BPEL, тогда как BPEL является языком описания оркестровки).

Несмотря на то, что эти термины повсеместно используются в технической литературе, посвященной концепциям BPM (Business Process Management, BPM) и SOA (сервис-ориентированная архитектура), отмечается базовых определений недостаточно для комплексного понимания этих архитектурных шаблонов. Рассмотрим оркестровку и хореографию с точки зрения их современного понимания научным и техническим сообществом, целей, ставящихся перед ними и задач, которые они решают<sup>1</sup>.

В [2] приводится абстрактная схема оркестровки и хореографии, которые двумя шаблонами описания бизнес-процессов в качестве взаимодействующих служб: в виде последовательного потока выполнения сервисов и в виде правил их взаимодействия (см. рис. 1).

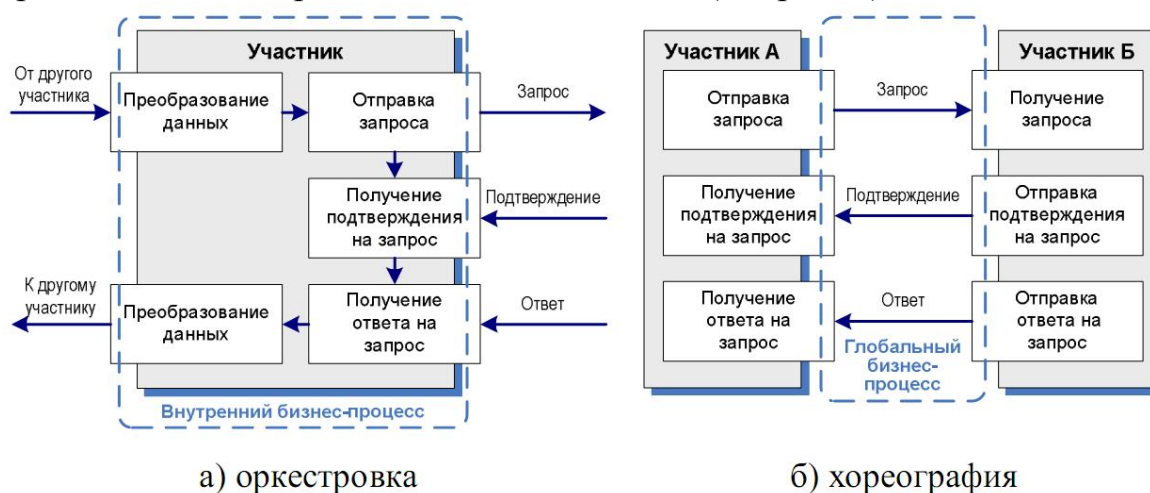


рис. 1 Схема оркестровки и хореографии [2]

Исходя из [1] и официального стандарта консорциума W3C [10] можно дать определение оркестровки как «описание внутреннего бизнес-процесса (сервиса) предприятия в виде потока и условий взаимодействия между внутренними и внешними для организации веб-сервисами (процессами)». При оркестровке существует некий центральный процесс, который управляет вызванными службами и операциями для достижения определенной цели. Вовлеченные во взаимодействие службы не знают, что

<sup>1</sup> Отметим, что ввиду определенной ограниченности языков и стандартов оркестровки и хореографии, они были рассмотрены лишь в качестве примеров, доказывающих те или иные обсуждаемые аспекты.

они были вызваны как часть от бизнес-процесса более высокого уровня. Оркестровка отличается явными описаниями операций и порядком вызова служб [2].

Для описания оркестровки с целью автоматизации процесса посредством системы управления бизнес-процессами или специализированного «движка» оркестровки используются языки моделирования, называемые языками оркестровки. К этим языкам можно отнести, например, BPEL, XPDL и BPMN [1].

На основании [2] и [10] можно дать определение хореографии как «последовательности и условий, при соблюдении которых несколько независимых участников обмениваются сообщениями с целью выполнения некоторой общей бизнес-задачи или достижения определенного целевого состояния». Хореография устанавливает участников, описывает их связи и способы управления композитным сервисом. В хореографии не предполагает центрального координатора, наоборот, каждая служба, вовлеченная в хореографию, знает точно, когда и с кем нужно взаимодействовать. При этом каждый участник хореографии должен знать обо всех выполняемых бизнес-процессах, операциях, сообщениях и настройках для их обмена. Раскрывается только видимое извне поведение каждого из участников, но не описываются никакие внутренние операции, которые происходят в рамках отдельных служб и которые напрямую не производят видимый для внешней среды эффект (например, такие операции, как внутренние вычисления или преобразование данных). Зачастую хореографию связывают с B2B взаимодействием, выводя независимость участников взаимодействия за пределы одной организации. К языкам описания хореографии относят WS-CDL (Web Services Choreography Description Language), ebXML Business Process Specification Schema (BPSS) и др.

Иногда в целях моделирования процессов хореографии и оркестровки и формализации таких моделей между этими шаблонами вводят промежуточный объект – поведенческий интерфейс<sup>2</sup> [16]. Интерфейс описывает поведение службы с точки зрения зависимостей от других служб. Выделяя роль службы, описывая взаимодействия в которые она вовлечена, типы и политики информационного обмена, интерфейс схож с хореографией, однако проводит описание с точки зрения локальной службы, чем схож с оркестровкой. Но, как и хореография, не затрагивает внутреннюю реализацию системы. Прототип поведенческого интерфейса

---

<sup>2</sup> Исходя из того, что поведенческий интерфейс играет лишь вспомогательную роль в процессах, касающихся оркестровки и хореографии, и является переходным шагом от сценариев хореографии к сценариям оркестровки, мы не будем его обсуждать в рамках этой статьи.

называется в BPEL «абстрактным процессом» и служит для описания хореографии в среде этого языка.

Некоторые авторы (например, в [8]) описывают, что отличие оркестровки и хореографии в самом физическом расположении взаимодействующих компонентов: оркестровка обеспечивается с локальными сервисами, а хореография – распределенными по разным предприятиям подсистемами. Однако концепция открытых стандартов, веб-служб и возможность распределенной работы таких систем контраргументирует этот довод: веб-сервис предназначен для работы в распределенной среде, через интернет и общедоступные, открытые протоколы. Современные крупные многопрофильные организации – это территориально распределенные комплексы производственных отделов и офисов, в рамках которых оркестровка может организовать взаимодействие ведущего процесса как с локальными подсистемами, так и с удаленными. Поэтому нельзя уверенно говорить о том, что хореография, в отличие от оркестровки, это шаблон, описывающий работу распределенных служб, так как обе эти концепции одинаково успешно могут справляться с композицией удаленных сервисов.

Так авторы ([4], [19]) описывают хореографию как поток сообщений между независимыми участниками. Но обмен сообщениями присущ и оркестровке, и хореографии. При оркестровке это сообщения перемещаются между внутренними и внешними сервисами, при хореографии описываются только внешние по отношению к службам сообщения. Однако сообщения – это лишь способ взаимодействия между сервисами, а цель хореографии – это организация этого взаимодействия путем определения условий, ролей и последовательностей взаимодействия. Налаженное взаимодействие требует обмена сообщениями, но это только одно из условий контрактов, устанавливаемых хореографией.

Базовое отличие, вытекающее из классического определения этих шаблонов – контроль над последовательностью взаимодействия. Оркестровка регламентирует работу одного процесса (службы) и описывает порядок вызова других процессов и служб с передачей данных и их преобразованием между вызовами. Взаимодействие рассматривается как движение «сверху вниз» [15] - передача управления от главного процесса-координатора к процессам-сателлитам, выполняющим вспомогательные функции. При этом иногда производится «жесткая» оркестровка служб, созданных в рамках одной и тоже модели программирования, и в этом случае службы объединяются программным кодом в новый композитный сервис. Таким образом, сценарий оркестровки выражается на определенном языке программирования. И поэтому можно отметить, что оркестровка демонстрирует логику кода императивных

языков программирования с использованием функций и процедур: последовательный вызов функций с параметрами и обработка полученных от них результатов. Такой подход прост и интуитивно понятен, много десятилетий реализуется различными языками программирования и поэтому основная нотация оркестровки – язык BPEL, является, по сути, языком программирования со всеми присущими свойствами.

Хореография же, наоборот, описывает не порядок взаимодействия, а условия этого взаимодействия для возможности передачи сообщений между участниками. Существует два принципиальных подхода к организации хореографии: подход, основанный на сообщениях и многоагентный подход. Многоагентный подход основан на контроле над поведением участников взаимодействия. Хореографией определяются различные, существенные стороны поведения каждого объекта, их назначения и роли, условия начала и завершения обмена сообщениями, условия перехода к следующему шагу. Такой подход частично поддерживает, например, язык BPEL, который позволяет описывать абстрактные процессы, и, избегая деталей выполнения, сосредотачиваться на отдельных аспектах поведения объектов. В случае подхода на сообщениях поведение глобального процесса определяется контролем над передаваемыми между участниками данными. Этот механизм поддерживается языком WS-CDL и часто используется для B2B взаимодействия. В B2B-приложениях, которые по определению распределены между предприятиями, сложно определить конкретную реализацию каждого участника и нет централизованной «власти» над всем потоком взаимодействия. Основанная на сообщениях хореография более удобна и прозрачна для реализации, так как необходимо только определить стандарты обмена сообщениями. С одной стороны, такой подход лишает участников хореографии определенной толики независимости, жестко регламентируя их роли и процессы обмена данными, с другой стороны, агентный подход может как дать участникам широкие возможности реализации, так и ограничить эту свободу серьезными правилами поведения. Так, в хореографии взаимодействие определяется поведением входящих компонентов, что соответствует движению «снизу вверх» [15]. Поэтому в некотором роде хореография идеологически схожа с декларативными языками программирования – описывает то, каково должно быть взаимодействие, а не то, как оно должно быть конкретно реализовано.

Из этих отличий оркестровки и хореографии следует еще одно: различается точка зрения, прилагаемая ко всему процессу целиком. Оркестровка рассматривает процесс-координатор в качестве владельца, ответственного за результаты своей работы. Именно он отвечает за

обработку ошибок, преобразование данных, проверку условий, разделение потока вычислений, его объединение и т.д. В хореографии же не требуется централизованного взгляда на весь процесс. Достаточно определить параметры поведения входящих в нее участников или характеристики обмена данными между ними.

В идеологии сервис-ориентированной архитектуры оркестровка играет одну из ведущих ролей. Именно об оркестровке идет речь, когда происходит интеграция слабо-связных сервисов низкого уровня в тот или иной бизнес-процесс (или очередной сервис) более высокого уровня. Здесь ведущий бизнес-процесс – это всего лишь оркестровка вспомогательных процессов более низкого уровня. В качестве индивидуальных «шагов» такой последовательности потока взаимодействия могут использоваться отдельные операции, реализуемые службами. Кроме того, основной индустриальный стандарт оркестровки BPEL использующий модели XML использует то же WSDL-описание служб на двух уровнях [15]: во-первых, описанные в терминах WSDL службы используются для взаимодействия в рамках процесса, а во-вторых, каждый BPEL-процесс – это такой же сервис, описанный на WSDL, соответствует стандартам сервисной архитектуры.

На первый взгляд, хореографии сложно найти применение в рамках SOA, но вместе с тем она хорошо вписывается в принципы таких архитектур систем, например, как ориентированных на события (англ.: event-driven) и на агентов<sup>3</sup> (англ.: agent-based).

Управляемая событиями система – архитектурный шаблон, который предполагает создание, обнаружение и реакцию на определенные события. Под событием понимается значимое изменение состояния определенного объекта. Например, когда покупается автомобиль в автосалоне, его состояние меняется с «на продаже» на «продан». Информационная система автосалона может воспринять изменение этого состояния как событие, которое должно быть обнаружено и обработано входящими в состав архитектуры приложениями. Так, на продажу может отреагировать система ведения бухгалтерского учета, процессы подготовки документов на автомобиль, система, обслуживающая гарантийную автомастерскую и пр.

---

<sup>3</sup> Заметим, что многоагентный подход уже упоминался в этой статье как подход к построению хореографии, однако в последнем случае речь идет об определенной архитектуре распределенных программных систем, где, кроме упомянутых event-driven- и agent-based-систем существует множество других шаблонов, как, например, описано в [11].

Многоагентная архитектура поддерживает существование и методы взаимодействия агентов в процессе функционирования системы. Агент – программная система, связанная со средой посредническими отношениями. Хореография может описать правила взаимодействия агентов или обмена сообщениями при возникновении определенного события в контексте системы. И вот эти архитектурные шаблоны уже, в свою очередь, хорошо подходят для поддержки сервис-ориентированной архитектуры. Передачу сообщений и информации о событиях, поддержку обмена данными между агентами может осуществлять на базе, например, шины ESB. Правила маршрутизации позволяют в ответ на событие запустить каскад реакций системы в виде последовательного или параллельного запуска отдельных сервисов или целых бизнес-процессов, каждый из которых может обладать определенной долей автономности и проявлять поведение программного агента или вовлекать в свою деятельность пользователей системы.

Свойство процессов и служб в СОА объединяться в крупные процессы и системы для реализации более сложной функциональности с точки зрения оркестровки и хореографии называют рекурсивной композицией [4]. Рекурсивная композиция предполагает, что сценарии оркестровки или хореографии могут комбинироваться с другими описаниями в новые сценарии оркестровки и хореографии, которые могут использоваться в отличном от изначального контексте. Оркестровка, представляющая с определенной точки зрения целостную службу или процесс, по природе своей поддерживает подобную композицию. Представление крупного процесса с точки зрения скомибинированных оркестровок – это представление его в виде «белого» ящика, где каждый включенный во взаимодействие процесс – это в первую очередь определенный сценарий. Хореография, наоборот, часто представляется исследователям в виде описания взаимодействия между частями целостной системы, и вопрос о рекурсивной композиции в таких случаях, конечно, не рассматривается. Сложность возникает при определении точки зрения на композитную хореографию для каждого из сценариев. С точки зрения вовлекаемого во взаимодействие сценария, то для него - композитный процесс более высокого уровня, а другие процессы, вовлекаемые в композитный – равны ему. Такой же вопрос можно задать по отношению к процессам, вовлекаемых в исходный сценарий и т.д. Этот вопрос можно решить, например, приняв решение о взаимодействии исключительно между сценариями хореографии, находящимися на одной уровне детализации. Каждый сценарий, как независимый участник, получает определенную роль в процессе композиции и выполняет свои задачи в рамках этой роли также как выполняют свои задачи уже вовлеченные в него процессы

(подобный пример можно отметить, например, для WS-CDL в [20]). Таким образом, невозможно одновременно создать побочную линию композиции хореографии в одном комбинированном сценарии с интеграцией хореографий, находящихся изначально на разных уровнях композиционной иерархии.

Присущая сценариям оркестровки схожесть с синтаксисом процедурных языков программирования влечет и схожие проблемы: затруднено описание сложных, многокомпонентных, длительных бизнес-процессов. Оркестровка вместе с языками ее реализации легка в изучении и позволяет быстро создавать простые сценарии взаимодействия служб, однако такие сценарии с трудом поддаются последующему масштабированию – логика вызовов сервисов и обработки данных пропорционально усложняется. Именно поэтому и в связи с необходимостью привлечения к разработке процессов специалистов нетехнического профиля [18] для сложных оркестровок были созданы инструменты визуализации потока исполнения, связанные, в основном, с графическим стандартом BPMN [1], и позволяющие генерировать исполняемый BPEL-код, который в свою очередь загружается в специализированную среду исполнения BPEL/BPM-движка. Намного сложнее представить и разработать средства графического моделирования хореографии. Частично, если сконцентрироваться на передаче сообщений между независимыми процессами, можно описать последовательность такой передачи, однако эта задача запросто решается средствами визуализации оркестровки или стандартными средствами того же языка UML. Сложнее, например, визуально описать правила взаимодействия независимых подсистем или скоординировать правила передачи таких сообщений. В этом случае, построение графического представления хореографии легко сравнимо с методами реверс-инжиниринга: по заданным условиям хореографии и изучению поведения отдельных участников взаимодействия можно построить соответствующую схему.

В отличие от оркестровки, процессы хореографии и поддерживающие ее стандарты сложнее в изучении и первоначальном применении. Однако за счет присущих возможностей широкого применения ранее созданных правил, регламентов и условий поведения и обмена сообщениями сценарии хореографии легко масштабируемы в крупные системы, сложность поведения которых определяется не столько хореографией, сколько взаимодействием участников.

Процесс оркестровки часто связан с наличием некоторой «информационной среды». Информационная среда представляет собой совокупность получаемой, хранимой и обрабатываемой координатором



процесса информацией вместе с процедурами ее обработки и проверки. Среда может быть выделена как для группы действий оркестровки, так и для всего процесса. Поэтому в процессе оркестровки можно определить свои переменные, проводить с ними несложные процедуры, проверять их при циклах и ветвлениях и т.д. В хореографии нет исполнителя центрального процесса, нет отдельного потока управления, поэтому создавать и поддерживать информационную среду для процесса хореографии можно только силами всех ее участников. Работа с информационными объектами ведется исключительно в границах процессов-участников, а проверка условий перехода к следующему шагу хореографии может проводить инициатор этого шага. Именно поэтому в средствах описания хореографии особая роль уделяется поддержке процессов информационного обмена и управления состоянием информационной составляющей каждого участника.

Исполняемый BPEL-код оркестровки действительно является «исполняемым» и схож со скриптовыми языками программирования: загружается в BPEL/BPM систему и исполняется как любой программный код через последовательный доступ к операциям веб-служб, манипуляцию с данными или ветвление/повторение потока управления и др. При этом «оркестрированный» поток отображается как самостоятельный сервис, взаимодействие с которым возможно через специфическое API. Сценарий хореографии не может быть исполнен напрямую, он «принимается к сведению» участниками хореографии, и позволяет на своей основе конфигурировать службы для взаимодействия и строить локальные сценарии оркестровки для каждой из них.

В этой статье были рассмотрены основные подходы к организации бизнес-процессов как взаимодействующих служб для сервисно-ориентированной архитектуры. Выявлены точки различий между оркестровкой и хореографией, как в теоретическом, так и в прикладном аспекте.

В оркестровке нашла отражение классическая методология программирования и описания потока работ: последовательное изменение состояния системы и включающих ее компонент для достижения определенного результата. Оркестровка регламентирует передачу управления под контролем центрального координатора, который является владельцем и ответственным за весь процесс. Как было показано, оркестровка более практична, легка в освоении и применении, однако слабо подходит для описания сложных взаимодействий независимых систем или для описания процессов высокого уровня абстракции. Хореография, напротив, отстраняясь от описания самого процесса, вводит правила и условия взаимодействия его участников. Именно поведение

участников и потоки информации между ними определяют роль глобального процесса.

Сервис-ориентированная архитектура и сервис-ориентированное программирование развили идеи объектно-ориентированного программирования и компонентно-ориентированного проектирования, выведя независимость программных объектов на новый уровень: отстраненность не только от текущего контекста (как в объектно-ориентированном программировании), но и от модели и среды разработки (как в компонентно-ориентированном программировании). Однако процессы взаимодействия независимых слабо-связных служб остаются описываемыми в рамках классических потоков работ, когда лишённые состояния объекты связываются в службы и вводятся каждый раз в новый целевой и бизнес-контекст. Это, по сути, традиционное повторное использование объекта, но на новом уровне.

Если оркестровка описывает путь достижения определенного целевого состояния системы, то хореография описывает само состояние. Хореография – возможно, более перспективный архитектурный шаблон и, возможно, это новый уровень взаимодействия независимых объектов, более высокий, чем предложено идеологией СОА. Каждый объект хореографии – независимый агент, обладающий собственными целями и поведением. Как и любой программный агент, он способен вступать во взаимодействие с другими объектами для достижения общей цели. В условиях независимости действий агента от человека и его свойствах проактивности хореография может стать тем ключом, который приведет к построению самосвязывающихся, самоорганизующихся систем. В которых не нужно будет описывать последовательность потока выполнения определенной работы, необходимо будет описывать правила взаимодействия и передачи сообщений, условия достижимости конечной цели и предоставлять участникам право самостоятельно выбирать пути достижения этой цели.

### **Список литературы**

1. Артамонов. И.В. Современные стандарты описания и исполнения бизнес-процессов / И.В. Артамонов // Применение математических методов и информационных технологий в экономике. – 2010. – С. 5 – 31.
2. Самуйлов К.Е. Бизнес-процессы и информационные технологии в управлении телекоммуникационными компаниями / К.Е. Самуйлов, А.В. Чукарин, Н.В. Яркина. – М.: Альпина Паблишерз, 2009. – 442 стр.

3. Aarti Karande et al. Choreography and Orchestration using Business Process Execution Language for SOA with Web Services / Aarti Karande, Milind Karande, B.B.Meshram. - IJCSI International Journal of Computer Science Issues - Vol. 8, Issue 2, 2011. - 224-232 стр.
4. Alistair Barros et al. A Critical Overview of the Web Services Choreography Description Language (WS-CDL)/ [электронный ресурс] // <http://www.bptrends.com/publicationfiles/03-05%20WP%20WS-CDL%20Barros%20et%20al.pdf>
5. Alistair Barros et al. Standards for Web Service Choreography and Orchestration: Status and Perspectives / Alistair Barros, Marlon Dumas, Phillipa Oaks / [электронный ресурс] // <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.3021>
6. Architectural Patterns and Styles / [электронный ресурс] // <http://msdn.microsoft.com/en-us/library/ee658117.aspx>
7. Changai Sun, Marco Aiello. Requirements and Evaluation of Protocols and Tools for Transaction Management in Service Centric Systems / [электронный ресурс] // [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?tp=&arnumber=4291162](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=4291162)
8. Charlton Barreto. What is... choreography? What is... orchestration? / [электронный ресурс] // [http://charltonb.typepad.com/weblog/2007/05/what\\_is\\_choreog.html](http://charltonb.typepad.com/weblog/2007/05/what_is_choreog.html)
9. Chris Peltz. Web Services Orchestration and Choreography / [электронный ресурс] // <http://portal.acm.org/citation.cfm?id=1437007>
10. David Booth and et al. Web Services Architecture. / [электронный ресурс] // <http://www.w3.org/TR/ws-arch/>
11. Distinction between "Choreography" and "Orchestration" / [электронный ресурс] // <http://tech.groups.yahoo.com/group/service-orientated-architecture/message/11347>
12. Dragos A. Manolescu. Patterns for Orchestration Environments / [электронный ресурс] // <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.141.5765>
13. Gustavo Ansaldi Oliva et al. Web Services Choreographies Adaptation: A Systematic Review, 2011 / Gustavo Ansaldi Oliva, Fernando Hattori, Leonardo Alexandre Ferreira Leite. Marco Aurélio Gerosa / [электронный ресурс]
14. Mike Havey. Essential Business Process Modeling / Mike Havey. - O'Reilly, 2005. - 350 стр.
15. Mike Rosen. BPM and SOA / [электронный ресурс] // [http://www.bptrends.com/publicationfiles/04-08-COL-BPMandSOA-OrchestrationorChoreography-%200804-Rosen%20v01%20\\_MR\\_final.doc.pdf](http://www.bptrends.com/publicationfiles/04-08-COL-BPMandSOA-OrchestrationorChoreography-%200804-Rosen%20v01%20_MR_final.doc.pdf)

16. Nickolas Kavantzias and et al. Web Services Choreography Description Language Version 1.0. / [электронный ресурс] // <http://www.w3.org/TR/ws-cdl-10/>
17. Oliver Kopp. The Influence of an External Transaction on a BPEL Scope. 2009. / [электронный ресурс] // <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.154.9338>
18. Raman Kazhamiakin and Marco Pistore. Choreography Conformance Analysis: Asynchronous Communications and Information Alignment / [электронный ресурс] // [http://klase.fbk.eu/paper\\_firb\\_astro\\_klase/additional/RK\\_wsfm06.pdf](http://klase.fbk.eu/paper_firb_astro_klase/additional/RK_wsfm06.pdf)
19. Ravi Khadka. Model-Driven Development of Service Compositions: Transformation from Service Choreography to Service Orchestrations / [электронный ресурс] // <http://essay.utwente.nl/59677/>
20. Web Services Business Process Execution Language Version 2.0. 2007 / [электронный ресурс] // <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>