

<b>A context-aware recommendation-based system for service composition in smart environments</b>	<b>Система контекстно-зависимых рекомендаций для организации услуг в интеллектуальных средах</b>
Soufiane Faieq <sup>1,2</sup> · Agnès Front <sup>1</sup> · Rajaa Saidi <sup>2,3</sup> · Hamid El Ghazi <sup>4</sup> · Moulay Driss Rahmani <sup>2</sup>	
Received: 18 June 2019 / Revised: 19 October 2019 / Accepted: 21 October 2019 © Springer-Verlag London Ltd., part of Springer Nature 2019	
<b>Abstract</b>	<b>Аннотация</b>
<p>The strong integration of technology in the physical world caused the emergence of smart environments. These environments are supposed to improve the quality of life of their users by providing them with customized services when needed and adapting to their changing needs. The dynamics of the users, the huge number of available services and the strong collaboration between the stakeholders, make traditional service-oriented approaches incapable of providing relevant services to the users in these environments. To deal with these issues, we propose a recommendation-based system for service composition targeting smart environments. The proposed system is able to capture the situation of the users through the analysis of their context information, which in turn allows the system to capture their requirements and select the appropriate service models to satisfy their needs. Then, based on the invocation log, the system implements two recommendation policies. First, it selects the best services in terms of QoS that satisfy the captured requirements. Second, the system recommends new tasks to be integrated in existing service models. The conducted experiments show the efficiency and effectiveness of the recommendation policies proposed. To illustrate the workings of the proposed system, we present a case study called SMARTROAD pertaining to the transport domain and road security.</p>	<p>Сильная интеграция технологий в физическом мире привела к появлению умных сред. Предполагается, что эти среды улучшат качество жизни своих пользователей, предоставляя им индивидуальные услуги по мере необходимости и адаптируясь к их меняющимся потребностям. Динамика пользователей, огромное количество доступных услуг и тесное сотрудничество между заинтересованными сторонами делают традиционные ориентированные на услуги подходы неспособными предоставлять пользователям соответствующие услуги в этих средах. Для решения этих проблем мы предлагаем основанную на рекомендациях систему составления услуг, ориентированную на интеллектуальные среды. Предлагаемая система способна фиксировать ситуацию пользователей посредством анализа их контекстной информации, что, в свою очередь, позволяет системе фиксировать их требования и выбирать соответствующие модели обслуживания для удовлетворения их потребностей. Затем на основе журнала вызовов система реализует две политики рекомендаций. Во-первых, он выбирает лучшие услуги с точки зрения QoS, которые удовлетворяют захваченным требованиям. Во-вторых, система рекомендует новые задачи для интеграции в существующие модели обслуживания. Проведенные эксперименты показывают эффективность и действенность предложенных рекомендаций. Чтобы проиллюстрировать работу предлагаемой системы, мы представляем тематическое исследование под названием SMARTROAD, относящееся к транспортной области и</p>

	безопасности дорожного движения.
<b>Keywords</b> Smart environment • Service composition • Service recommendation • Context awareness • Quality of service prediction • Service model	<b>Ключевые слова</b> Умная среда, Состав сервисов, Сервисная рекомендация, Контекстная осведомленность, Прогноз качества обслуживания, Сервисная модель
<b>1 Introduction</b>	1 Введение
<p>The world of Information and Communication Technology and Computer Science holds great potential and promise in the transformation of today's big cities toward smart cities. The smart city has been defined as "...an innovative city that uses information and communication technologies (ICTs) and other means to improve quality of life, efficiency of urban operations and services, and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social and environmental aspects" [1]. Harnessing ICT advances is the new way to deal with the issues related to today's big cities (e.g., resource consumption), caused especially by the increasing number of residents within those cities [2]. This strong integration of technology in the physical world caused the emergence of "smart environments." We define these environments as <i>virtual extensions of the physical world that are capable of providing customized services to their users when needed and adapting them to their changing needs</i>. Hence, smart environments are characterized by (i) being dynamic, heterogeneous and mobile, (ii) being dependent on the advances of technology, (iii) having a user-centered vision and (iv) providing a myriad of services.</p>	<p>Мир информационных и коммуникационных технологий и компьютерных наук обладает огромным потенциалом и перспективами в преобразовании современных крупных городов в умные города. Умный город был определен как «... инновационный город, который использует информационные и коммуникационные технологии (ИКТ) и другие средства для улучшения качества жизни, эффективности городских операций и услуг, а также конкурентоспособности, обеспечивая при этом, чтобы это отвечало потребностям современных и будущих поколений в отношении экономических, социальных и экологических аспектов» [1]. Использование достижений в области ИКТ является новым способом решения проблем, связанных с современными крупными городами (например, потреблением ресурсов), вызванных, в частности, увеличением числа жителей этих городов [2]. Эта сильная интеграция технологий в физическом мире привела к появлению «умных сред». Мы определяем эти среды как виртуальные расширения физического мира, которые способны предоставлять индивидуальные услуги своим пользователям при необходимости и адаптировать их к их меняющимся потребностям. Поэтому интеллектуальные среды характеризуются тем, что (i) они являются динамичными, разнородными и мобильными, (ii) зависят от достижений технологии, (iii) имеют ориентированное на пользователя видение и (iv) предоставлять множество услуг.</p>
<p>In smart environments, providing services that are relevant to the users is a necessity. There are different types of services that are and will be offered to the citizens in different application domains (e.g., transport, energy, home, health, etc.), by different service providers in the context</p>	<p>В интеллектуальных средах предоставление услуг, актуальных для пользователей, является необходимостью. Существуют различные типы услуг, которые предлагаются и будут предлагаться гражданам в различных областях применения (например, транспорт,</p>

of smart cities, and their enabling technologies (i.e., data storage services, data processing services, sensing services, etc.) [3]. It comes naturally to think about service-oriented computing (SOC) when evoking the notion of services. Service-oriented computing has proven itself as the go-to paradigm for designing, delivering and consuming services, mainly through the realization of service-oriented architectures (SOA) [4]. Using SOA as a guideline allows the creation of distributed software systems that are loosely coupled, flexible and platform agnostic, mostly based on the principles of reuse and composition of several services that are modular, autonomous, loosely coupled and self-describing and that can be offered internally or through third parties. A popular example technology implementing SOA is Web services. Web services can be defined as Web applications that are able to interact and exchange messages via standard application-to-application protocols over well-defined interfaces. The interaction of more than two Web services is known as “Web Service Composition (WSC)” [5] and aims to develop value-added services while reducing development, integration and maintenance costs and time to market via increasing reuse, sharing and flexibility.

Meanwhile, data science and its related disciplines (e.g., machine learning, data mining, etc.) are expected to play a major role in the “smartification” movement [6]. It is used to develop systems that are capable of delivering actionable information to other systems or to decision makers, through the analysis of the huge amount of data transiting on the network nowadays. Recommender systems are a

энергетика, бытовая техника, здравоохранение и т.д.) различными поставщиками услуг в контексте «умных городов» и их технологиями поддержки (т.е. услуги хранения данных, услуги обработки данных, услуги зондирования и т. д.) [3]. Естественно думать о сервисно-ориентированных вычислениях (SOC), когда речь идет об услугах. Сервисно-ориентированные вычисления зарекомендовали себя как парадигма перехода к разработке, предоставлению и потреблению сервисов, в основном за счет реализации сервис-ориентированных архитектур (SOA) [4]. Использование SOA в качестве руководства позволяет создавать распределенные программные системы, которые являются слабосвязанными, гибкими и независимыми от платформы, основанными на принципах повторного использования и составления нескольких сервисов, которые являются модульными, автономными, слабосвязанными и имеют самоописание и которые могут предлагаться внутри или через третьих лиц. Популярным примером технологии, реализующей SOA, являются веб-сервисы. Веб-службы могут быть определены как веб-приложения, которые могут взаимодействовать и обмениваться сообщениями через стандартные протоколы между приложениями через четко определенные интерфейсы. Взаимодействие более двух веб-сервисов известно, как «Композиция веб-сервисов (WSC)» [5] и направлено на разработку дополнительных услуг при одновременном снижении затрат на разработку, интеграцию и обслуживание, а также времени на вывод на рынок за счет увеличения повторного и совместного использования, а также гибкости.

Между тем, наука о данных и связанные с ней дисциплины (например, машинное обучение, интеллектуальный анализ данных и т. Д.), как ожидается, будут играть важную роль в движении «умелости» [6]. Они используются для разработки систем, способных доставлять полезную информацию другим системам или лицам, принимающим решения, посредством анализа огромного

<p>popular implementation of such systems, as they produce relevant items for the active users based on the analysis of previously recorded actions performed by past users. Recommender systems have shown their effectiveness and have been used for several application domains such as, tourism, e-commerce, news and entertainment. The use of recommender systems in the development of service-oriented software solutions can be beneficial in several aspects, especially those where we need to get actionable information from incomplete and sparse data (e.g., recommending services based on incomplete quality of service (QoS) data) [7].</p>	<p>количества данных, проходящих в сети в настоящее время. Рекомендательные системы являются популярной реализацией таких систем, поскольку они производят соответствующие элементы для активных пользователей на основе анализа ранее записанных действий, выполненных предыдущими пользователями. Рекомендательные системы показали свою эффективность и использовались для нескольких областей применения, таких как туризм, электронная коммерция, новости и развлечения. Использование рекомендательных систем при разработке сервисно-ориентированных программных решений может быть полезным в нескольких аспектах, особенно в тех, где нам необходимо получать полезную информацию из неполных и разреженных данных (например, рекомендации услуг на основе данных неполного качества обслуживания (QoS)) [7].</p>
<p>The characteristics of smart environments mentioned above in the first paragraph make service composition in these environments challenging. On the one hand, the huge number of services available makes it harder to select relevant services for the user [8]. The difficulty here lies, not only in selecting the best service to perform a particular task (i.e., in terms of QoS), but also in analyzing the situation of the user and reacting to it while minimizing the interactions with the user. On the other hand, smart environments (e.g., smart city) are based on the close and strong collaboration between different actors, blurring the lines between the different domains. This makes it hard for domain experts and developers to create, maintain and update collaboration scenarios that are later translated into service models.</p>	<p>Характеристики интеллектуальных сред, упомянутые выше в первом абзаце, усложняют состав услуг в этих средах. С одной стороны, огромное количество доступных услуг затрудняет выбор соответствующих услуг для пользователя [8]. Трудность здесь заключается не только в выборе наилучшего сервиса для выполнения конкретной задачи (то есть в терминах QoS), но также в анализе ситуации пользователя и реагировании на него при минимизации взаимодействия с пользователем. С другой стороны, интеллектуальные среды (например, «умный город») основаны на близком и прочном сотрудничестве между различными участниками, стирая грани между различными областями. Это затрудняет создание и поддержание и обновление сценариев совместной работы для экспертов в области предметной области и разработчиков, которые впоследствии преобразуются в модели обслуживания.</p>
<p>To address the challenges discussed above, we explore situational awareness using the notion of context to model and capture the situation of the user. Knowing the situation of the user is helpful for delivering the relevant services to the particular user in a transparent manner. We also</p>	<p>Для решения проблем, обсуждаемых выше, мы исследуем ситуационную осведомленность, используя понятие контекста для моделирования и отображения ситуации, в которой находится пользователь. Знание ситуации пользователя полезно для</p>

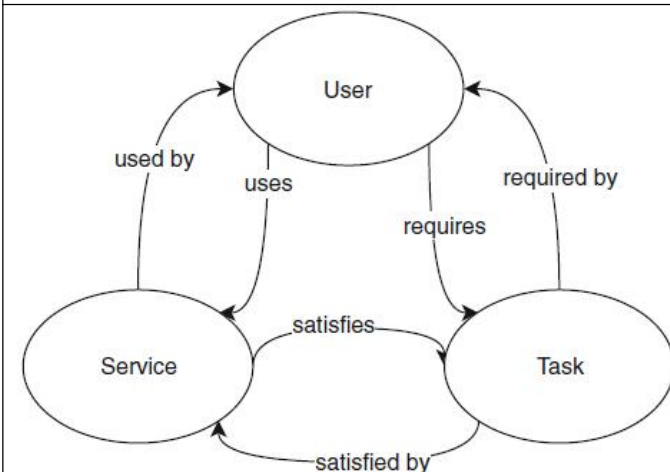
<p>explore the use of recommendation techniques to address the challenges at two levels: First, in selecting the best services in terms of non-functional properties for each user (i.e., QoS properties), second, in assisting the developers and domain experts in the maintenance and updates of the service models, through recommending new tasks to be integrated.</p>	<p>прозрачного предоставления соответствующих услуг конкретному пользователю. Мы также исследуем использование методов рекомендаций для решения проблем на двух уровнях: во-первых, при выборе лучших услуг с точки зрения нефункциональных свойств для каждого пользователя (т. Е. Свойств QoS), во-вторых, при помощи разработчиков и экспертов в области поддержание и обновление сервисных моделей путем рекомендации новых задач для интеграции.</p>
<p>In this paper, we propose a context-aware system for service composition based on recommendation. The system captures and processes context information to deduce the situation of its users. Based on the deduced situation, the appropriate service model that is capable of reacting to the needs of each user is selected for execution. To select the best services to satisfy each task in the service model, we use a collaborative filtering framework based on autoencoders to predict the QoS (i.e., QoS) values and then choose the service with the best predicted value. The system also interacts with the domain experts and developers to recommend relevant tasks for integration in the predefined service models. The SMARTROAD case study is presented and used throughout the paper to illustrate the working process of the system.</p>	<p>В этой статье мы предлагаем контекстно-зависимую систему для организации услуг на основе рекомендаций. Система собирает и обрабатывает контекстную информацию для определения ситуации своих пользователей. На основании выведенной ситуации для выполнения выбирается подходящая модель сервиса, способная реагировать на потребности каждого пользователя. Чтобы выбрать лучшие сервисы для удовлетворения каждой задачи в модели сервиса, мы используем систему совместной фильтрации, основанную на автокодировщиках, для прогнозирования значений QoS (т.е. QoS), а затем выбираем сервис с наилучшим прогнозируемым значением. Система также взаимодействует с экспертами и разработчиками предметной области, чтобы рекомендовать соответствующие задачи для интеграции в предопределенные модели обслуживания. Тематическое исследование SMARTROAD представлено и используется на протяжении всей статьи, чтобы проиллюстрировать рабочий процесс системы.</p>
<p>The rest of this paper is structured as follows. Section 2 introduces the relevant notions and concepts used throughout this paper. Section 3 introduces the motivating scenario behind our system, which is also used to illustrate its working process. The system, its components and processes are described in Sect. 4. Section 5 presents the results of the experiments conducted to evaluate the performance of the recommendation approaches on the data generated by the system. In Sect. 6, we discuss the novelty of our approach when compared to</p>	<p>Остальная часть этой статьи структурирована следующим образом. Раздел 2 вводит соответствующие понятия и концепции, используемые в этой статье. Раздел 3 представляет мотивирующий сценарий нашей системы, который также используется для иллюстрации ее рабочего процесса. Система, ее компоненты и процессы описаны в разд. 4. В разделе 5 представлены результаты экспериментов, проведенных для оценки эффективности рекомендательных подходов к данным, сгенерированным системой.</p>



existing systems in the literature. We finally conclude this paper and give some future research directions in Sect. 7.

## 2 Background

In this section, we give a small introduction into the main concepts that are in play in this paper. We first introduce service composition (SC) as the principle allowing to produce composite services from atomic (unit) services. Second, we present recommender systems (RS) as systems allowing to suggest interesting items to users based on historical records of implicit or explicit user feedback. Lastly, we talk about context awareness and its potential for capturing the situation of the users of the system and reacting to it by providing the users with what they need.



**Fig. 1** Interactions between the main concepts in SOC

### 2.1 Service composition

Service composition (SC) is one of the core design principles of service-oriented computing (SOC) and ensures that atomic services (i.e., simple

В разделе 6, мы обсуждаем новизну нашего подхода по сравнению с существующими в литературе системами. Мы наконец завершаем эту статью и даем некоторые дальнейшие направления исследований в разделе. 7.

## 2 Контекст

В этом разделе мы дадим небольшое введение в основные концепции, которые используются в этой статье. Сначала мы вводим состав услуг (SC) в качестве принципа, позволяющего производить составные услуги из атомарных (единичных) услуг. Во-вторых, мы представляем рекомендательные системы (RS) как системы, позволяющие предлагать пользователям интересные элементы на основе исторических записей неявных или явных отзывов пользователей. Наконец, мы говорим о контекстной осведомленности и ее возможностях для того, чтобы уловить ситуацию пользователей системы и реагировать на нее, предоставляя пользователям то, что им нужно.

Рис 1. Взаимодействие между основными понятиями в SOC

### 2.1. Состав сервисов

Композиция услуг (SC) является одним из основных принципов проектирования сервис-ориентированных вычислений (SOC) и

or unit services) can collaborate and be aggregated into one or more composite services (i.e., complex services) [5]. The service composition process involves several procedures, and these procedures in their turn involve other procedures. According to [9], this process is depicted as a life cycle consisting of four major steps: (1) *Definition*, which allows the translation of the request of the user toward an abstract process (i.e., service model); (2) *Selection*, which achieves the mappings between each task in the service model to a concrete service that is capable of performing the said task; (3) *Deployment*, which converts the abstract process to an executable process using the selected concrete services; (4) *Execution*, which is where the executable process is ran and can be monitored. Hence, there are several concepts and roles involved in typical SC approaches. Figure 1 depicts the interactions between the main actors in typical service-oriented solutions, where three concepts are in constant interaction, namely services, users and tasks.

Services usually perform specific tasks and can be classified into atomic services and composite services [10]. An atomic service is a single unit which can no longer be decomposed, while a composite service is the ordered or parallel execution of two or more services. Tasks can be considered as the functional part of the service. Consequently, it is natural that services follow tasks in their nature (i.e., atomic or composite). In SC, a task (atomic or composite) is also called a *service model* or an abstract service [11] because it represents the functional aspect of the service, ignoring its implementation details. Users can be developers who create the service models (composite tasks) by specifying the tasks that need to be performed and the order in which they are performed later on by the services, or final users who consume the available developed services to accomplish specific tasks.

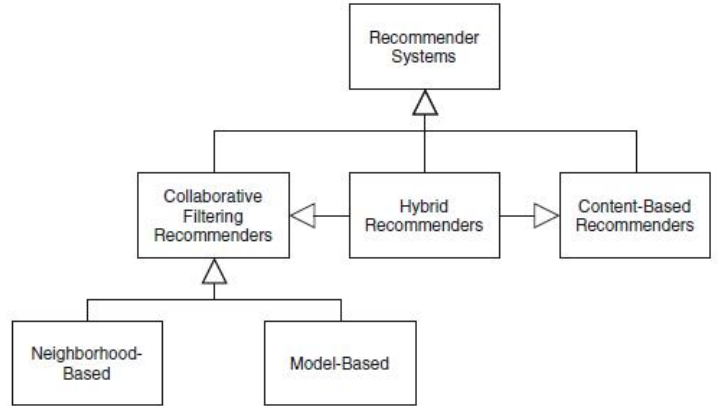
гарантирует, что элементарные сервисы (т. Е. Простые или единичные сервисы) могут сотрудничать и объединяться в одну или несколько составных сервисов (т. Е. Сложных сервисов). [5]. Процесс составления сервиса включает несколько процедур, и эти процедуры, в свою очередь, включают другие процедуры. Согласно [9] этот процесс изображается как жизненный цикл, состоящий из четырех основных этапов: (1) Определение, которое позволяет преобразовать запрос пользователя в абстрактный процесс (т. Е. Модель обслуживания); (2) Выбор, при котором достигается соответствие между каждой задачей в модели сервиса конкретной услуге, которая способна выполнить указанную задачу; (3) Развертывание, которое преобразует абстрактный процесс в исполняемый процесс, используя выбранные конкретные службы; (4) Выполнение, где исполняемый процесс запускается и может контролироваться. Следовательно, есть несколько концепций и ролей, вовлеченных в типичные подходы SC. На рисунке 1 изображено взаимодействие между основными действующими лицами в типичных сервис-ориентированных решениях, где три понятия находятся в постоянном взаимодействии, а именно услуги, пользователи и задачи.

Сервисы обычно выполняют конкретные задачи и могут быть классифицированы на атомарные сервисы и композитные сервисы [10]. Элементарная служба - это единичный модуль, который больше не может быть разложен, в то время как составная служба - это упорядоченное или параллельное выполнение двух или более служб. Задачи можно рассматривать как функциональную часть сервиса. Следовательно, естественно, что сервисы следуют задачам по своей природе (то есть атомарным или составным). В SC задача (элементарная или составная) также называется моделью службы или абстрактной службой [11], поскольку она представляет функциональный аспект службы, игнорируя детали ее реализации. Пользователи могут быть разработчиками, которые создают модели сервисов (составные задачи), определяя задачи, которые должны

	<p>быть выполнены, и порядок, в котором они выполняются позже сервисами, или конечными пользователями, которые используют доступные разработанные сервисы для выполнения конкретных задач.</p>
<p><b>2.2 Recommender systems</b></p>	<p>2.2 Рекомендательные системы</p>
<p>With the plethora of information available online today, users find themselves in a sea of irrelevant data depending on the task they are trying to achieve. Recommender systems (RS) came as away to address that problem and are highly adopted in e-commerce applications where there is a myriad of products and were also used in different information retrieval problems [12]. In recommender systems, the products to be recommended are generally called Items and these items are recommended to Users. In these systems, we assume that each user can evaluate an item through a Rating system, either explicitly (e.g., the user gives a number of stars on a maximum of five stars) or implicitly (e.g., browsed items in a session, number of clicks, etc.).</p>	<p>С огромным количеством информации, доступной сегодня в сети, пользователи оказываются в море несущественных данных в зависимости от задачи, которую они пытаются достичь. Рекомендательные системы (RS) появились для решения этой проблемы и широко используются в приложениях электронной коммерции, где существует множество продуктов, а также использовались для решения различных задач поиска информации [12]. В рекомендательных системах рекомендуемые продукты обычно называются Предметами, и эти предметы рекомендуются Пользователям. В этих системах мы предполагаем, что каждый пользователь может оценить элемент с помощью системы рейтинга, либо явно (например, пользователь дает количество звездочек максимум на пять звездочек), либо неявно (например, просмотренные элементы в сеансе, количество щелчки и т. д.).</p>
<p>Different approaches are used to implement recommender systems depending on the nature of the data describing the users, items and ratings (see Fig. 2). The most popular approach is <i>Collaborative Filtering</i> (CF) [13], where the general idea is to model the process of recommendation through collaboration in real life (i.e., Person A recommending to Person B a certain product or movie) to filter the items [14]. This is translated to systems as the assumption that people are more likely to be interested in items that are already liked by other people with a similar taste to their own. CF techniques can be generally classified into neighborhood-based techniques or model-based techniques. Neighborhood-based techniques use the neighborhood of the active user (i.e., users similar to the active user based on past ratings or likes or showing somewhat similar behavior) or the item (i.e., items similar to the items previously liked or highly rated by the active user) to recommend</p>	<p>Для реализации рекомендательных систем используются разные подходы в зависимости от характера данных, описывающих пользователей, предметы и рейтинги (см. Рис. 2). Наиболее популярным подходом является коллаборативная фильтрация (CF) [13], где общая идея состоит в том, чтобы смоделировать процесс рекомендации посредством совместной работы в реальной жизни (т. Е. Персона А рекомендует человеку В определенный продукт или фильм) для фильтрации элементов [14]. Это переводится на системы как предположение о том, что люди с большей вероятностью будут интересоваться предметами, которые уже нравятся другим людям, которые имеют вкус, похожий на их собственный. Методы CF обычно могут быть классифицированы на методы, основанные на соседстве, или методы, основанные на моделях. Методы, основанные на соседстве, используют соседство</p>



<p>interesting items. These types of recommenders scale linearly to the number of users and items [15]. Model-based techniques use the available information to build a model that is able to generate the recommendation. Most often, the data are transformed into a latent space model where the weights are learned through a training process. Among this family of techniques, methods based on matrix factorization [16] have particularly attracted much attention after they showed good efficiency at the Netflix Prize competition.<sup>1</sup> However, other methods based on neural networks [17], genetic algorithms [18], latent features and graph theory [19] are also widely used.</p>	<p>активного пользователя (то есть пользователей, похожих на активного пользователя на основе прошлых оценок или лайков или демонстрирующих несколько похожее поведение) или элемент (т. Е. Элементы, похожие на элементы, ранее понравившиеся или высоко оцененные активным пользователем), чтобы рекомендовать интересные предметы. Эти типы рекомендаций масштабируются линейно в зависимости от количества пользователей и предметов [15]. Основанные на модели методы используют доступную информацию для построения модели, способной генерировать рекомендации. Чаще всего данные преобразуются в модель скрытого пространства, где веса изучаются в процессе обучения. Среди этого семейства методов методы, основанные на матричной факторизации [16], особенно привлекли большое внимание после того, как они показали хорошую эффективность на конкурсе Netflix Prize. Однако другие методы, основанные на нейронных сетях [17], генетические алгоритмы [18], латентные особенности и теория графов [19] также широко используются.</p>
<p><i>Content-based recommenders</i> adopt a different concept. They recommend items that are similar or related to the ones that the user liked (or highly rated) in the past. This similarity is usually computed based on the features of the items alone [20]. Thus, this kind of recommenders is based on the assumption that the items can easily be described into categorical data types. This requires a strong domain knowledge which is not always given and can be hard to maintain. For example, movies can be described by their genres, cast, director, year of production, etc. Another drawback of these types of systems is serendipity, which is translated by their inability to recommend new items that might interest the user but that are not similar or related to previously rated items.</p>	<p>Контент-ориентированные рекомендации принимают другую концепцию. Они рекомендуют элементы, которые похожи или связаны с теми, которые понравились пользователю (или высоко оцененные) в прошлом. Это сходство обычно рассчитывается на основе характеристик только предметов [20]. Таким образом, этот тип рекомендаций основан на предположении, что элементы могут быть легко описаны в категориальных типах данных. Это требует сильного знания предметной области, которое не всегда дается и может быть трудно поддерживать. Например, фильмы можно описать по их жанрам, актерам, режиссерам, годам выпуска и т. Д. Другим недостатком этих типов систем является случайность, что выражается в их неспособности рекомендовать новые элементы, которые могут заинтересовать пользователя, но не похожие или связанные с ранее оцененными предметами.</p>



**Fig. 2** Recommender systems: types and techniques

*Hybrid recommenders* can be any combination of the previously mentioned recommenders using any combination of techniques [21]. The goal is to leverage the strength of each type of recommender to address or mitigate the drawbacks of another.

**2.3 Context awareness**

Context information has been defined by Abowd and Dey [22] as “...any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. This definition inspired the concept of context-aware computing which is centered around the ability of a system to collect, analyze and react to the changes in the context. This ability makes systems capable of making smarter decisions and offering more tailored and personalized content, services and experiences.

Due to the excessive amount of information available nowadays, context information can be found everywhere. This information can be of several types, representing different things. For example, location, as a popular context type, can be expressed by an address or by latitude and longitude

**Рис 2.** Рекомендательные системы: типы и методы

Гибридные рекомендации могут представлять собой любую комбинацию ранее упомянутых рекомендаций с использованием любой комбинации методов [21]. Цель состоит в том, чтобы усилить силу каждого типа рекомендации для устранения или смягчения недостатков другого.

**2.3. Контекстная осведомленность**

Контекстная информация была определена Abowd and Dey [22] как «... любая информация, которая может использоваться для характеристики ситуации сущности. Сущность - это человек, место или объект, которые считаются относящимися к взаимодействию между пользователем и приложением, включая пользователя и сами приложения». Это определение вдохновило концепцию контекстно-зависимых вычислений, которая основана на способности системы собирать, анализировать и реагировать на изменения в контексте. Эта способность делает системы способными принимать более разумные решения и предлагать более индивидуальный и персонализированный контент, услуги и опыт.

Из-за чрезмерного количества информации, доступной в настоящее время, контекстную информацию можно найти повсюду. Эта информация может быть нескольких типов, представляющих разные вещи. Например, местоположение, как популярный тип

<p>coordinates. However, some types of context information can have more impact and significance than others, depending on the application. Hence, it is necessary to analyze the impact of each type of contextual information on the performance of the systems to be able to leverage contextual awareness effectively and efficiently [23]. Focusing on the context information that brings the most value also reduces the amount of data to be stored and processed which reflects positively on the performance of context-aware systems.</p>	<p>контекста, может быть выражено адресом или координатами широты и долготы. Однако некоторые типы контекстной информации могут иметь большее влияние и значимость, чем другие, в зависимости от приложения. Следовательно, необходимо проанализировать влияние каждого типа контекстной информации на производительность систем, чтобы иметь возможность эффективно и результативно использовать контекстную осведомленность [23]. Сосредоточение внимания на контекстной информации, которая приносит наибольшую пользу, также уменьшает объем данных, которые должны храниться и обрабатываться, что положительно сказывается на производительности систем с учетом контекста.</p>
<p><b>3 Case study: SMARTROAD</b></p>	<p><b>3 Пример использования: SMARTROAD</b></p>
<p>In this section, we introduce a case study to get a better understanding of the problem and the objectives of this paper.</p>	<p>В этом разделе мы представляем тематическое исследование, чтобы лучше понять проблему и цели этой статьи.</p>
<p>There are 1.35 million human deaths related to traffic accidents worldwide. According to the World Health Organization, the majority of this number is registered in Africa [24]. In Morocco, for example, the road safety situation has been described as a “road war,” with close to 82,000 accidents in 2016, an increase of 3.8% when compared to 2015, which is mainly due to the rise of the number in vehicles and automobile industry in the country [25]. This amount of accidents has effects on the economy, environment, society and mobility, and a similar situation is also noticed in the rest of the Arab Maghreb Union and developing countries in general. In this regard, Morocco had launched the national strategy for road safety 2016–2025, aiming at halving the number of deaths in road accidents.</p>	<p>Во всем мире в результате дорожно-транспортных происшествий погибли 1,35 миллиона человек. По данным Всемирной организации здравоохранения, большая часть этого числа зарегистрирована в Африке [24]. Например, в Марокко ситуация с безопасностью дорожного движения была описана как «дорожная война», в которой в 2016 году было почти 82 000 дорожно-транспортных происшествий, что на 3,8% больше по сравнению с 2015 годом, что в основном связано с увеличением числа транспортных средств и автомобильная промышленность в стране [25]. Такое количество несчастных случаев оказывает влияние на экономику, окружающую среду, общество и мобильность, и аналогичная ситуация наблюдается и в остальной части Союза арабского Магриба и в развивающихся странах в целом. В связи с этим Марокко приступило к осуществлению национальной стратегии обеспечения безопасности дорожного движения на 2016–2025 годы, направленной на сокращение вдвое числа смертей в результате дорожно-транспортных происшествий.</p>
<p>Road safety, however, is only one of the issues related to the surface</p>	<p>Однако безопасность дорожного движения является лишь одной из</p>

<p>transport landscape. In fact, transport is at the heart of any nation's continuous and sustainable development. From an organizational perspective, different stakeholders are involved and each one has a particular perspective in the planning, building, managing, using and analyzing of the transport landscape (e.g., Policymakers, vehicle suppliers, energy providers, service providers, endcustomers, etc.). These stakeholders are actors that influence the transport ecosystem in mainly four levels: (1) infrastructure, (2) vehicle, (3) driver and (4) environment. Any effort to reduce the number of accidents and deaths on the road has to deal with the pre-crash phase and the post-crash phase with respect to the different actors in the transport ecosystem. In this paper, we focus on the pre-crash phase, where the goal is to manage and minimize the risk factors related to each actor using the services provided by the different stakeholders.</p>	<p>проблем, связанных с ландшафтом наземного транспорта. На самом деле транспорт является основой непрерывного и устойчивого развития любой страны. С организационной точки зрения участвуют различные заинтересованные стороны, и каждый из них имеет особую перспективу в планировании, создании, управлении, использовании и анализе транспортной среды (например, политики, поставщики транспортных средств, поставщики энергии, поставщики услуг, конечные клиенты и т.д.), эти заинтересованные стороны являются субъектами, которые влияют на транспортную экосистему в основном на четырех уровнях: (1) инфраструктура, (2) транспортное средство, (3) водитель и (4) окружающая среда. Любые усилия по сокращению количества аварий и смертей на дороге должны быть связаны с фазой до аварии и фазой после аварии в отношении различных участников транспортной экосистемы. В этой статье мы фокусируемся на этапе, предшествующем сбою, где цель состоит в том, чтобы управлять и минимизировать факторы риска, связанные с каждым участником, используя услуги, предоставляемые различными заинтересованными сторонами.</p>
<p>In this paper, we are interested in the services provided to the end-customers by the service providers and city authorities. Table 1 presents some examples of the services that can be offered to the smart city inhabitants, their service provider and their supporting entity [26,27].</p>	<p>В этой статье мы заинтересованы в услугах, предоставляемых конечным клиентам поставщиками услуг и городскими властями. В таблице 1 представлены некоторые примеры услуг, которые могут быть предложены жителям умного города, их поставщику услуг и организации поддержки [26, 27].</p>
<p>To effectively deal with the potentially problematic situations (e.g., traffic, bad weather, etc.) that may arise on the roads and their potential causes (In this paper, we call them <i>Risk Factors</i>), several of the services presented in Table 1 have to be composed. We have focused on road safety by defining situations that can be considered dangerous and the risk factors that can contribute to the occurrence of these dangerous situations. We consider that each risk factor can cause a dangerous situation. These risk factors are related to the different entities involved in the road transport ecosystem (i.e., vehicle, driver, infrastructure and</p>	<p>Чтобы эффективно справляться с потенциально проблемными ситуациями (например, с дорожным движением, плохой погодой и т. Д.), которые могут возникнуть на дорогах, и их потенциальными причинами (в этом документе мы называем их факторами риска), некоторые из служб, представленных в таблице 1, <b>должны быть составлены</b>. Мы сосредоточились на безопасности дорожного движения, определяя ситуации, которые можно считать опасными, и факторы риска, которые могут способствовать возникновению этих опасных ситуаций. Мы считаем, что каждый фактор риска</p>

environment) and can generally be monitored through sensors that can be accessed via exposed services (e.g., speed, driver health and weather).

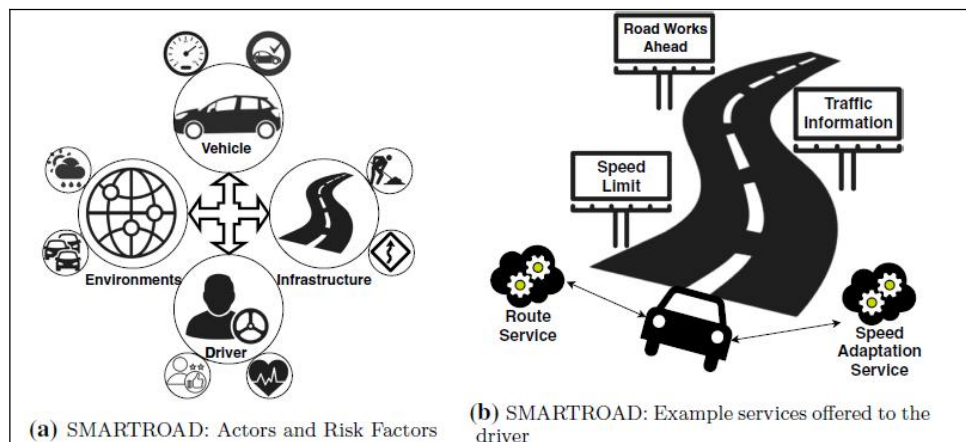
может вызвать опасную ситуацию. Эти факторы риска связаны с различными объектами, участвующими в экосистеме автомобильного транспорта (т.е. с транспортным средством, водителем, инфраструктурой и окружающей средой), и, как правило, могут контролироваться с помощью датчиков, к которым можно получить доступ через доступные службы (например, скорость, состояние здоровья водителя и погода)

**Table 1** Examples of provided Services

Table 1 Examples of provided services	Acronym	Service Name	Stakeholder
	AEVW	Approaching Emergency Vehicle Warning	Emergency Services
	CBW	Car Breakdown Warning	Automobile Industry
	RMS	Road Monitoring Service	Road Operators
	VSMS	Vehicle State Monitoring Service	Automobile Industry
	ISA	Intelligent Speed Adaptation	Automobile Industry
	IVS	In-Vehicle Signage	Automobile Industry
	TJAW	Traffic Jam Ahead Warning	Road Operators
	ERP	Electronic Road Panel Service	Road Operators
	RWW	Road Works Warning	Road Operators
	WWS	Weather Warning Service	Weather Services
	SMSD	Short Messaging Service for Driver	Automobile Industry
	RSS	Route Selection Service	Automobile Industry
	ERS	Emergency Rescue Service	Emergency Services
	NAAS	Nearby Area Alarming Service	Road Operators
	TSS	Towing Selection Service	Navigation Services
	OCS	Oil Calculation Service	Automobile Industry
	GSS	Garage Selection Service	Navigation Services
	GSSS	Gas Station Selection Service	Navigation Services
	HSS	Hospital Selection Service	Health-care Services

Таблица 1. Примеры предоставляемых услуг





**Fig. 3** SMARTROAD: actors, risk factors and services

Figure 3a shows these entities and their related risk factors, as defined in this case study.

We linked each situation to a composite service model (composition model) to inform the entities that can be influenced by each given situation. The aim is to make these entities aware of the dangers and to enable the authorities to take the necessary steps to return to a normal state of traffic. Table 2 presents each risk factor, the name of the situation and the service model to mitigate its dangers.

Like almost all smart city applications, the case study presented in this subsection involves the participation of many actors (stakeholders and decision makers), and their exposed services need to work together to achieve the true connectedness of the smart city. In this paper, we divide the actors into final users (drivers) and professionals (transport specialists with software knowledge or developers). In service-based applications, final user's are usually ignorant to the logic of the application. However, the common practice is to allow them to choose the suitable services to perform specific tasks when it is possible, especially when there is no time constraint on the choice being taken

Рис 3. SMARTROAD: действующие лица, факторы риска и услуги

На рисунке 3а показаны эти объекты и связанные с ними факторы риска, как это определено в данном тематическом исследовании.

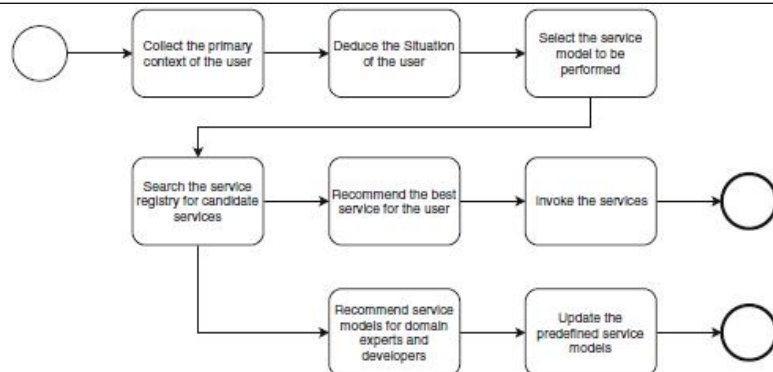
Мы связали каждую ситуацию с моделью составного сервиса (составной моделью), чтобы проинформировать объекты, на которые может влиять каждая конкретная ситуация. Цель состоит в том, чтобы информировать эти организации об опасностях и дать властям возможность предпринять необходимые шаги, чтобы вернуться к нормальному состоянию движения. В таблице 2 представлен каждый фактор риска, название ситуации и модель обслуживания для смягчения ее опасностей.

Как и почти во всех приложениях для «умного города», тематическое исследование, представленное в этом подразделе, включает в себя участие многих действующих лиц (заинтересованных сторон и лиц, принимающих решения), и предоставляемые ими услуги должны работать вместе для достижения истинной связности «умного города». В этой статье мы делим участников на конечных пользователей (водителей) и специалистов (транспортные специалисты со знаниями в области программного обеспечения или разработчики). В сервисных приложениях конечные пользователи обычно не знают логику

(e.g., Garage Selection Service).	приложения. Однако общепринятой практикой является предоставление им возможности выбирать подходящие услуги для выполнения конкретных задач, когда это возможно, особенно когда нет никаких временных ограничений на выбор, который делается (например, служба выбора гаража).										
<b>4 Recommendation-based system for service Composition</b>	4 Основанная на рекомендациях система определения состава услуг										
In this section, we first present the design of our recommendation- based system for service composition and its components through the system architecture and a component diagram. Then, we move on to explain in more details the workings of each component.	В этом разделе мы сначала представим проект нашей системы рекомендаций по составу услуг и ее компонентам через архитектуру системы и диаграмму компонентов. Затем мы переходим к более подробному объяснению работы каждого компонента.										
<p><b>Table 2</b> List of possible situations and their triggered compositions</p> <table border="1"> <thead> <tr> <th>Situation Name</th><th>Triggered Composition</th></tr> </thead> <tbody> <tr> <td>RiskyDriverHealth</td><td> <pre> graph LR   ISA --&gt; ERS   ISA --&gt; RSS   ISA --&gt; NAAS   ERS --&gt; AEVW   AEVW --&gt; ERP   RSS --&gt; IVS   NAAS --&gt; IVS </pre> </td></tr> <tr> <td>RiskyVehicleState</td><td> <pre> graph LR   VSMS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; GSS   NAAS --&gt; IVS   GSS --&gt; RSS   RSS --&gt; IVS </pre> </td></tr> <tr> <td>RiskyRoad</td><td> <pre> graph LR   RMS --&gt; ISA   RMS --&gt; RWW   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS   RWW --&gt; SMSD </pre> </td></tr> <tr> <td>BadWeather</td><td> <pre> graph LR   WWS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS </pre> </td></tr> </tbody> </table>	Situation Name	Triggered Composition	RiskyDriverHealth	<pre> graph LR   ISA --&gt; ERS   ISA --&gt; RSS   ISA --&gt; NAAS   ERS --&gt; AEVW   AEVW --&gt; ERP   RSS --&gt; IVS   NAAS --&gt; IVS </pre>	RiskyVehicleState	<pre> graph LR   VSMS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; GSS   NAAS --&gt; IVS   GSS --&gt; RSS   RSS --&gt; IVS </pre>	RiskyRoad	<pre> graph LR   RMS --&gt; ISA   RMS --&gt; RWW   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS   RWW --&gt; SMSD </pre>	BadWeather	<pre> graph LR   WWS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS </pre>	<p>Таблица 2 Список возможных ситуаций и вариантов их срабатываний</p>
Situation Name	Triggered Composition										
RiskyDriverHealth	<pre> graph LR   ISA --&gt; ERS   ISA --&gt; RSS   ISA --&gt; NAAS   ERS --&gt; AEVW   AEVW --&gt; ERP   RSS --&gt; IVS   NAAS --&gt; IVS </pre>										
RiskyVehicleState	<pre> graph LR   VSMS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; GSS   NAAS --&gt; IVS   GSS --&gt; RSS   RSS --&gt; IVS </pre>										
RiskyRoad	<pre> graph LR   RMS --&gt; ISA   RMS --&gt; RWW   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS   RWW --&gt; SMSD </pre>										
BadWeather	<pre> graph LR   WWS --&gt; ISA   ISA --&gt; NAAS   ISA --&gt; RSS   NAAS --&gt; ERP   RSS --&gt; IVS </pre>										
<b>4.1 System overview</b>	4.1 Обзор системы										

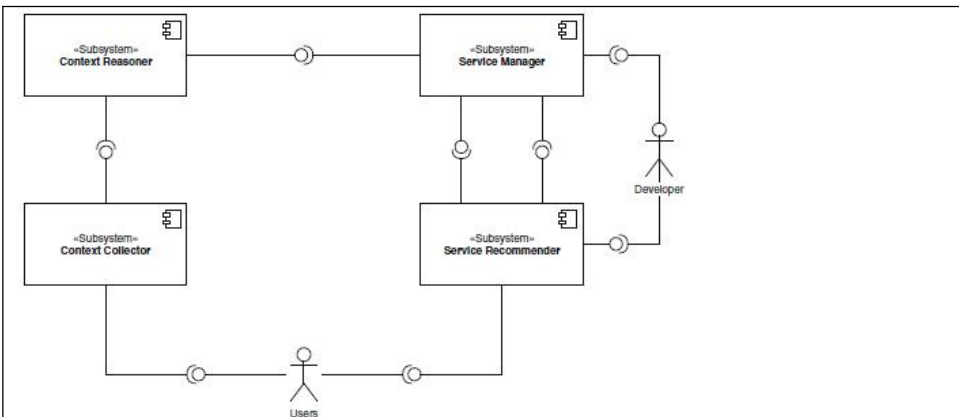
As mentioned in Sect. 1, the proliferation of services in smart environments has made service selection a challenging task. On the one hand, the selected services should be tailored to the need of the user and deliver the best QoS possible at execution time. On the other hand, the high collaboration between the different stakeholders in these environments makes it difficult for the service developers and domain experts to conceive service models that are capable of satisfying the specific requirements of each user. To address these challenges, we propose in this paper a recommender-based system for context-aware service composition. The goals of this system are threefold: 1. Analyze the context of the user to personalize the provided services to his particular situation and needs 2. Simplify the collaboration procedures between the different stakeholders to define the service models through recommending the tasks to be integrated 3. Offer the best services in terms of quality to each user through analyzing previous invocation data

Как упомянуто в разд. 1, распространение услуг в интеллектуальных средах сделало выбор услуг сложной задачей. С одной стороны, выбранные сервисы должны быть адаптированы к потребностям пользователя и обеспечивать наилучшее качество обслуживания во время выполнения. С другой стороны, высокая степень взаимодействия между различными заинтересованными сторонами в этих средах затрудняет разработчикам услуг и экспертам в области разработку моделей услуг, способных удовлетворить конкретные требования каждого пользователя. Чтобы решить эти проблемы, в этой статье мы предлагаем систему на основе рекомендаций для контекстно-зависимой структуры услуг. Цели этой системы имеют три цели: 1. Анализировать контекст пользователя, чтобы персонализировать предоставляемые услуги в соответствии с его конкретной ситуацией и потребностями. 2. Упростить процедуры сотрудничества между различными заинтересованными сторонами, чтобы определить модели обслуживания посредством рекомендации задач для интеграции. 3 Предложить лучшие услуги с точки зрения качества каждому пользователю, анализируя данные предыдущего вызова.



**Fig. 4** Execution process of the recommender-based service composition system

**Рис. 4** Процесс выполнения в системе составления сервисов на основе рекомендаций



**Fig. 5** Recommender-based service composition system: component diagram

Figure 4 shows a process that illustrates the order in which each task of the system is executed. Figure 5 illustrates the actors and the software components of our proposed system. Figure 6 presents the system architecture when applied to the SMARTROAD case study. We chose the layers based on a Separation of Concerns principle [28]. The functions of each layer in the system architecture is detailed in the following subsections.

## 4.2 Context collection

This is the first step, and the bottom layer that is needed in every context-aware or knowledge-based information system or software. It provides the necessary methods and processes to extract the necessary data from the users of the system. As we stated before in Sect. 3, context is every bit of information that can help in identifying the situation of a user. The context factors to take into account are closely related to the application. In our case study, for example, the goal is to make the road secure. Hence, the context factors that are of interest are the ones that would allow the identification of there being a risk, the nature of the risk, the actors responsible for it and the actors that are affected by it. We denote by  $C = (c1, c2, \dots, cn)$  the set of context factors that are relevant to the

**Рис. 5** Система составления сервисов на основе рекомендаций: схема компонентов

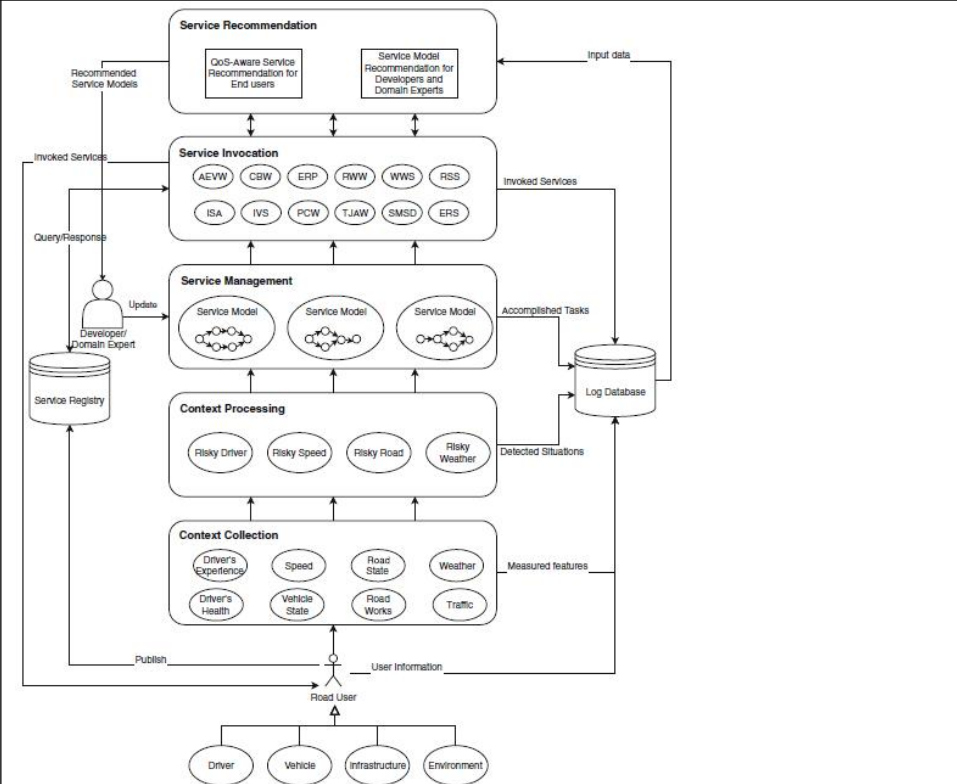
На рисунке 4 показан процесс, который иллюстрирует порядок, в котором выполняется каждая задача системы. На рисунке 5 показаны действующие лица и программные компоненты нашей предлагаемой системы. На рисунке 6 представлена архитектура системы применительно к примеру, SMARTROAD. Мы выбрали слои на основе принципа разделения интересов [28]. Функции каждого уровня в архитектуре системы подробно описаны в следующих подразделах.

## 4.2. Сбор контекста

Это первый шаг и нижний уровень, который необходим в каждой контекстно-зависимой или основанной на знаниях информационной системе или программном обеспечении. Он предоставляет необходимые методы и процессы для извлечения необходимых данных от пользователей системы. Как мы уже говорили ранее в разд. 3, контекст - это каждый бит информации, который может помочь в идентификации ситуации пользователя. Факторы контекста, которые необходимо учитывать, тесно связаны с приложением. Например, в нашем тематическом исследовании цель заключается в обеспечении безопасности дороги. Следовательно, контекстные факторы, представляющие интерес, - это те, которые

application and  $n$  the number of these factors.

позволят выявить наличие риска, природу риска, субъектов, ответственных за него, и субъектов, на которых он влияет. Обозначим через  $C = (c_1, c_2, \dots, c_n)$  набор контекстных факторов, которые имеют отношение к приложению, и  $n$  число этих факторов.



**Fig. 6** Recommender-based system architecture for the SMARTROAD prototype

Context factors can further be divided into two categories. The first category is the primary context, which consists of the set of context factors that are directly involved in the logic of the system (e.g., risk factors in our case study). The context factors in this category are a must have for the system to operate. The second category is the secondary context, which assembles the context factors that may enhance the

**Рис. 6** Архитектура системы на основе рекомендаций для прототипа SMARTROAD

Факторы контекста могут быть далее разделены на две категории. Первая категория - это основной контекст, который состоит из набора факторов контекста, которые непосредственно участвуют в логике системы (например, факторы риска в нашем тематическом исследовании). Контекстные факторы в этой категории являются обязательными для работы системы. Вторая категория представляет



<p>performance of the system in specific tasks (e.g., location of the user). The context factors in this category are not necessary for the system to function. However, they can significantly improve its performance when and if they are available and used effectively.</p>	<p>собой вторичный контекст, который объединяет факторы контекста, которые могут повысить производительность системы в конкретных задачах (например, местоположение пользователя). Контекстные факторы в этой категории не обязательны для функционирования системы. Тем не менее, они могут значительно улучшить его производительность, когда и если они доступны и используются эффективно.</p>
<p>In a previous work, we proposed a MOF [29] (Meta- Object Facility) compliant metamodel (presented in Fig. 7) to simplify the specification of contextual properties related to the users of the system [30]. This enables the classification of the contextual properties and answers the questions associated with the nature of each context property. At the same time, it allows an abstraction from specifying an exact format to transmit the collected data (e.g., key-value, xml, ontologies, etc.).</p>	<p>В предыдущей работе мы предложили метамодель, соответствующую MOF [29] (Meta-Object Facility) (представлена на рис. 7), чтобы упростить спецификацию контекстных свойств, связанных с пользователями системы [30]. Это позволяет классифицировать контекстные свойства и дает ответы на вопросы, связанные с природой каждого свойства контекста. В то же время он позволяет абстракции от указания точного формата для передачи собранных данных (например, значение ключа, xml, онтологии и т.д.).</p>
<p>In this paper, we assume that the needed context information is accessible through services provided by the actors in the system. In our case study, we consider each risk factor as a primary contextual factor that is monitored through the previously mentioned services (e.g., the speed of the vehicles, the health of the drivers, etc.).</p>	<p>В этой статье мы предполагаем, что необходимая контекстная информация доступна через сервисы, предоставляемые участниками системы. В нашем тематическом исследовании мы рассматриваем каждый фактор риска в качестве основного контекстуального фактора, который отслеживается с помощью ранее упомянутых услуг (например, скорость транспортных средств, состояние здоровья водителей и т. Д.).</p>

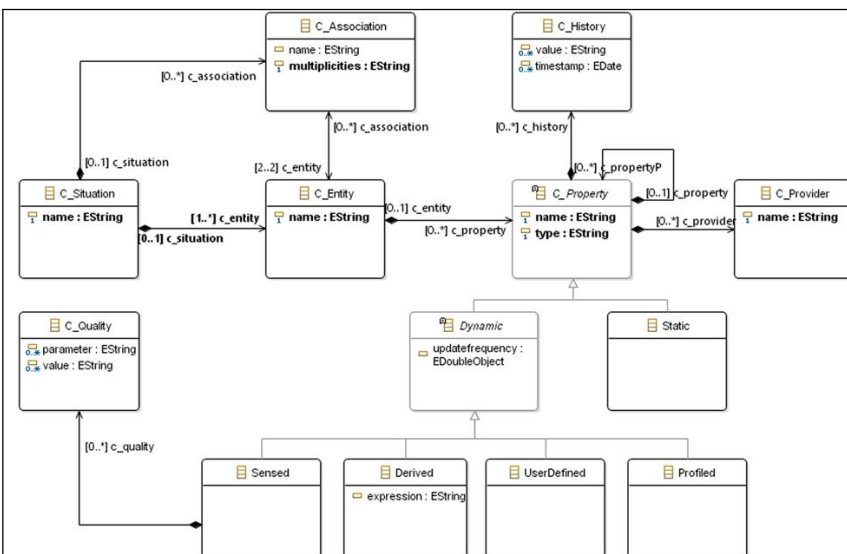


Fig. 7 MOF compliant context metamodel

Рис. 7. MOF-совместимая контекстная метамодель

### 4.3 Context processing

Context processing is the task of reasoning over the collected context information. The goal of this layer is to be able to identify the situation of one or more users of the system. To do that, we analyze the collected values of the context factors that are of interest in the application. The analysis consists on setting rules about the values of context factors that would allow an action to be performed upon the activation of a rule statement (e.g., *if Vehicle\_Speed  $\geq$  allowed\_speed then inform\_police*). Domain experts are usually tasked with defining these rules if the application domain is of a sensitive nature (e.g., Road security).

We periodically check the situation of the user through analyzing the different context factors specified at the modeling stage. After applying each specified rule on its corresponding context factor, the result is a one-hot vector of size  $n$ . We call this vector the *Status* of the user. An

### 4.3. Обработка контекста

Обработка контекста - это задача обдумывания собранной контекстной информации. Цель этого уровня - определить состояние одного или нескольких пользователей системы. Для этого мы анализируем собранные значения контекстных факторов, представляющих интерес для приложения. Анализ состоит в установке правил о значениях факторов контекста, которые позволили бы выполнить действие после активации оператора правила (например, если *Vehicle\_Speed  $\geq$  allow\_speed* затем *inform\_police*). Обычно эти правила поручают определять специалистам в предметной области, если предметная область приложения имеет особый характер (например, безопасность дорожного движения).

Мы периодически проверяем ситуацию пользователя, анализируя различные контекстные факторы, указанные на этапе моделирования. После применения каждого указанного правила к соответствующему контекстному фактору в результате получается

example of the resulting vector based on our case study is presented in Table 3. The example shows that the current driver is inexperienced, driving in a dangerous road and in a bad weather.

#### 4.4 Service model management

Service management encompasses the different tasks that are necessary to the design, storage and execution of orchestrations (i.e., servicemodels). The goal of this layer is to provide the developers and domain experts with the necessary tools to create the service models. Particularly, they need to specify the tasks to be performed, the order in which those tasks will be performed and any eventual data dependencies between the tasks.

This layer reacts to the signals sent by the context processing layer. Indeed, service models are designed to react specifically to each situation that is defined in the context processing layer. The service models are just abstractions of the needed concrete services. They only describe the general behavior of the services and the tasks that are to be performed. For simplicity, we defined a unique service model for each situation. In our case study, the list of situations and their corresponding service models are presented in Table 2.

**Table 3** User status example

Environment		Infrastructure		Vehicle		Driver	
Traffic	Weather	Condition	Safety	Speed	State	Experience	Health
0	1	1	0	0	0	1	0

#### 4.5 Service invocation

The service execution layer is responsible for the discovery, matching and consumption of the concrete services that are available in the registry. The resulting services need to satisfy one or multiple tasks in the service model. The goal here is to handle: (i) checking the

горячий вектор размера  $n$ . Мы называем этот вектор Статусом пользователя. Пример результирующего вектора, основанного на нашем тематическом исследовании, представлен в Таблице 3. Пример показывает, что текущий водитель неопытен, ездит по опасной дороге и в плохую погоду.

#### 4.4 Управление сервисной моделью

Управление сервисами включает в себя различные задачи, которые необходимы для проектирования, хранения и выполнения оркестровок (т.е. сервисные модели). Цель этого уровня - предоставить разработчикам и экспертам в области необходимые инструменты для создания моделей сервисов. В частности, им необходимо указать задачи, которые необходимо выполнить, порядок их выполнения и любые возможные зависимости данных между задачами.

Этот уровень реагирует на сигналы, посылаемые уровнем обработки контекста. Действительно, сервисные модели предназначены для конкретной реакции на каждую ситуацию, определенную на уровне обработки контекста. Сервисные модели - это просто абстракции необходимых конкретных сервисов. Они описывают только общее поведение служб и задачи, которые должны быть выполнены. Для простоты мы определили уникальную модель обслуживания для каждой ситуации. В нашем тематическом исследовании список ситуаций и соответствующие им сервисные модели представлены в таблице 2.

Таблица 3 Пример статуса пользователя

#### 4.5 Сервисный вызов

Слой выполнения службы отвечает за обнаружение, сопоставление и использование конкретных служб, доступных в реестре. Получающиеся сервисы должны удовлетворять одной или несколькими задачам в сервисной модели. Цель здесь состоит в том,

availability of the services, (ii) network access between the system and the available services and (iii) the data dependencies of the service (i.e., the inputs required by each service if any). As the service discovery [31] topic is out of the scope of this paper, we only use the name of the task to discover functionally relevant services in the service repository.	чтобы обрабатывать: (i) проверку доступности услуг, (ii) доступ к сети между системой и доступными услугами и (iii) зависимости данных от услуги (т.е. входные данные, необходимые для каждой услуги, если таковые имеются) ). Поскольку тема обнаружения сервисов [31] выходит за рамки данной статьи, мы используем имя задачи только для обнаружения функционально релевантных сервисов в репозитории сервисов.
<b>4.6 Service recommendation</b>	<b>4.6. Рекомендательные сервисы</b>
As mentioned in Sect. 2.2, recommendation systems are an important aspect of knowledge-based information systems. In service computing, service recommendation is a topic of great interest. The recommendations are generated based on one or several characteristics (e.g., time, location, QoS [32], trust, current activity, etc.). In this paper, we focus on two characteristics to generate service recommendations for two different users of the system.	Как упомянуто в разд. 2.2, системы рекомендаций являются важным аспектом информационных систем, основанных на знаниях. В сервисных вычислениях сервисная рекомендация является предметом большого интереса. Рекомендации генерируются на основе одной или нескольких характеристик (например, время, местоположение, QoS [32], доверие, текущая активность и т. Д.). В этой статье мы сфокусируемся на двух характеристиках, чтобы сформировать сервисные рекомендации для двух разных пользователей системы.
First, for the final users of the system, we generate the best services that are capable of satisfying a particular task. We base the recommendation on the QoS recorded by the system in previous invocations. Second, for the developers and domain experts, we recommend the tasks that may be integrated or added to the predefined service models. To do so, we analyze the services that were invoked by the users and whose service model (i.e., task) is not integrated in the predefined service models.	Во-первых, для конечных пользователей системы мы создаем лучшие сервисы, способные решить конкретную задачу. Мы основываем рекомендацию на QoS, зафиксированном системой в предыдущих вызовах. Во-вторых, для разработчиков и экспертов в области мы рекомендуем задачи, которые могут быть интегрированы или добавлены в предопределенные модели сервисов. Для этого мы анализируем сервисы, которые были вызваны пользователями и чья сервисная модель (то есть задача) не интегрирована в предопределенные модели сервисов.
<b>4.6.1 QoS-aware service recommendation for end users</b>	<b>4.6.1 QoS-осведомленный рекомендательный сервис, для конечных пользователей</b>
Smart and ubiquitous environments offer a big number of services potentially providing the same functionality to the users. QoS properties are used as non-functional parameters to select the best services. However, recorded QoS properties depend on the user, the service and their corresponding context. This makes the task of selecting the best	Умные и вездесущие среды предлагают большое количество сервисов, потенциально предоставляющих пользователям одинаковую функциональность. Свойства QoS используются в качестве нефункциональных параметров для выбора лучших услуг. Однако записанные свойства QoS зависят от пользователя, службы

<p>service for a particular user troublesome. Furthermore, the knowledge about QoS properties is not complete as that would imply that every user has invoked every available service. This in turn makes service selection even more challenging.</p>	<p>и соответствующего контекста. Это усложняет задачу выбора лучшего сервиса для конкретного пользователя. Кроме того, знание свойств QoS не является полным, поскольку это подразумевает, что каждый пользователь вызывал все доступные службы. Это, в свою очередь, делает выбор услуг еще более сложным.</p>
<p>The goal here is to provide the best service in terms of QoS to achieve each task in the service model. Knowing that a user only invokes a small set of the available services. This translates to a prediction problem where the goal is to predict the missing values of a matrix <math>R \in R_{nu \times ns}</math> where <math>nu</math> is the number of users and <math>ns</math> is the number of services. The rows of the matrix <math>R</math> represent the users of the system <math>U</math>, where each user <math>u \in U = (1, \dots, nu)</math> can be represented by a partially observed vector <math>r(u) = (Ru, 1, \dots, Ru, ns) \in R_{ns}</math>. The columns of <math>R</math> represent the services <math>S</math>, where each service <math>s \in S = (1, \dots, ns)</math> can be represented by a partially observed vector <math>r(s) = (Rs, 1, \dots, Rs, nu) \in R_{nu}</math>. Each cell in <math>c \in R</math> represents a QoS property value recorded when a user <math>u \in U</math> invokes a service <math>s \in S</math>.</p>	<p>Цель здесь - предоставить лучший сервис с точки зрения QoS для достижения каждой задачи в модели сервиса. Зная, что пользователь вызывает только небольшой набор доступных сервисов. Это приводит к проблеме прогнозирования, где цель состоит в том, чтобы предсказать недостающие значения матрицы <math>R \in R_{nu \times ns}</math>, где <math>nu</math> - это число пользователей, а <math>ns</math> - это количество услуг. Строки матрицы <math>R</math> представляют пользователей системы <math>U</math>, где каждый пользователь <math>u \in U = (1, \dots, nu)</math> может быть представлен частично наблюдаемым вектором <math>r(u) = (Ru, 1, \dots, Ru, ns) \in R_{ns}</math>. Столбцы <math>R</math> представляют сервисы <math>S</math>, где каждый сервис <math>s \in S = (1, \dots, ns)</math> может быть представлен частично наблюдаемым вектором <math>r(s) = (Rs, 1, \dots, Rs, nu) \in R_{nu}</math>. Каждая ячейка в <math>c \in R</math> представляет значение свойства QoS, записанное, когда пользователь <math>u \in U</math> вызывает службу <math>s \in S</math>.</p>
<p>The challenges here are mainly related to the nature of the data generated by the service. In this paper, we focused on QoS data which are objective, meaning that contrary to classic recommendation problems where the data are subjective and hence reflect the user's satisfaction with an item (i.e., movie, music, news, etc.); data in QoS service recommendation data are objective and hence only represent objective factors about the services for it a certain user. Basically, that means that unlike recommendations over subjective data where identical user is more likely to give a close ratings for the same item, it is highly possible that the identical users have varying ratings for the same service in QoS data.</p>	<p>Проблемы здесь в основном связаны с характером данных, генерируемых службой. В этой статье мы сфокусировались на данных QoS, которые являются объективными, что означает, что в отличие от классических проблем с рекомендациями, когда данные являются субъективными и, следовательно, отражают удовлетворение пользователя предметом (например, фильм, музыка, новости и т.д.); данные в рекомендациях обслуживания QoS являются объективными и, следовательно, представляют собой только объективные факторы об услугах для определенного пользователя. По сути, это означает, что в отличие от рекомендаций по субъективным данным, когда идентичный пользователь с большей вероятностью будет давать близкие оценки для одного и того же элемента, весьма вероятно, что идентичные пользователи</p>



<p>To achieve this task, we adopted the framework proposed by [33] called AutoRec, which is a collaboration filtering framework based on autoencoders. In their approach, they aim to design an autoencoder which can take as input a set of partially observed <math>r(s)</math> or <math>r(u)</math>, project it into a lowdimensional latent space and then reconstruct it in the output space to predict missing ratings. They formally defined the problem as, given a set <math>X</math> of vectors in <math>\mathbb{R}^d</math>, and some <math>k \in \mathbb{N}^+</math>, an autoencoder solves</p>	<p>имеют разные оценки для одного и того же сервиса в данных QoS.</p> <p>Чтобы решить эту задачу, мы приняли предложенную [33] структуру под названием AutoRec, которая представляет собой среду фильтрации совместной работы, основанную на автокодировщиках. В своем подходе они стремятся спроектировать автокодировщик, который может принимать в качестве входных данных набор частично наблюдаемых <math>r(s)</math> или <math>r(u)</math>, проецировать его в скрытое пространство меньшей размерности, а затем реконструировать его в выходном пространстве для прогнозирования недостающих оценок. Они формально определили задачу следующим образом: для заданного множества <math>X</math> векторов in <math>\mathbb{R}^d</math> и некоторого <math>k \in \mathbb{N}^+</math> автокодировщик решает</p>
$\min_{\theta} \sum_{r \in S} \ r - h(r; \theta)\ _2^2,$	
<p>where <math>h(r; \theta)</math> is the reconstruction of input <math>r \in \mathbb{R}^d</math>,</p>	<p>где <math>h(r; \theta)</math> - реконструкция входа <math>r \in \mathbb{R}^d</math>,</p>
$h(r; \theta) = f(W \cdot g(Vr + \mu) + b)$	
<p><math>f(\cdot)</math> and <math>g(\cdot)</math> are activation functions. Here, <math>\theta = \{W; V; \mu; b\}</math>, where <math>V \in \mathbb{R}^{k \times d}</math> and <math>W \in \mathbb{R}^{d \times k}</math> are the weight matrices for encoding (i.e., projecting) and decoding (i.e., reconstructing) the input, respectively, and <math>\mu \in \mathbb{R}^k</math> and <math>b \in \mathbb{R}^d</math> are biases. This objective corresponds to an auto-associative neural network [34], with a single hidden layer (HL) with <math>k</math> dimension. The parameters <math>\theta</math> are learned using backpropagation. To avoid overfitting, we regularize the learned parameters by introducing the regularization term <math>R = \ W\ _F^2 + \ V\ _F^2</math> and use a regularization rate parameter <math>\lambda</math> to control the strength of the regularization.</p>	<p><math>f(\cdot)</math> и <math>g(\cdot)</math> - функции активации. Вот, <math>\theta = \{W; V; \mu; b\}</math>, где <math>V \in \mathbb{R}^k \times d</math> и <math>W \in \mathbb{R}^d \times k</math> - весовые матрицы для кодирования (то есть проецирования) и декодирования (то есть восстановления) соответственно ввода, а <math>\mu \in \mathbb{R}^k</math> и <math>b \in \mathbb{R}^d</math> являются смещениями. Эта цель соответствует автоассоциативной нейронной сети [34] с одним скрытым слоем (HL) с <math>k</math>-мерностью. Параметры <math>\theta</math> изучаются с использованием обратного распространения. Чтобы избежать переобучения, мы упорядочиваем изученные параметры, вводя член регуляризации <math>R = \ W\ _F^2 + \ V\ _F^2</math> и используем параметр скорости регуляризации <math>\lambda</math> для управления силой регуляризации.</p>
<p>To the best of our knowledge, the application of this approach in the service computing domain was never attempted. Hence, this paper can be considered a stepping stone toward the integration and application of more deep learning frameworks in service computing.</p>	<p>Насколько нам известно, применение этого подхода в области вычислительных услуг никогда не предпринималось. Следовательно, этот документ можно считать ступенькой к интеграции и применению более глубоких структур обучения в сервисных вычислениях.</p>
<p><b>4.6.2 Service model recommendation for developers and domain experts</b></p>	<p>4.6.2. Модель рекомендательного сервиса для разработчиков и экспертов в предметной области</p>

<p>Nowadays, the changes in business processes are quite frequent due to changing market, policies, misconceptions, new trends, etc. Domain experts and developers cannot keep up with these changes. This exposes the need for recommendations about the tasks to be integrated in the predefined service models and business processes.</p>	<p>В настоящее время изменения в бизнес-процессах происходят довольно часто из-за изменения рынка, политики, неправильных представлений, новых тенденций и т.д. Эксперты в предметной области и разработчики не могут идти в ногу с этими изменениями. Это обуславливает необходимость рекомендаций по задачам для интеграции в предопределенные модели обслуживания и бизнес-процессы.</p>
<p>The goal here is to inform domain experts and developers about the need for potential changes in the predefined service models. Even further, we aim to provide them with the potential elements to be integrated into that change. To do so, we analyze the execution traces (i.e., <i>Events</i>) that were recorded to discover new patterns in the data. We proceed first by filtering the traces to extract the tasks which do not belong to any predefined service model (i.e., <i>UniqueAtomicTasks</i>, where <i>UAT</i> which means <i>UniqueAtomicTask</i> is an element of <i>UniqueAtomicTasks</i>). In the mean time, we group the atomic events together by task (i.e., <i>Grouped AtomicEvents</i>, where <i>GAE</i> which means <i>Grouped Atomic Event</i> is an element of <i>Grouped AtomicEvents</i>). Second, we extract the set of users who needed the tasks of the invoked services (i.e., <i>Users</i>, where <i>U</i> which means <i>User</i> is an element of <i>Users</i>). Third, we analyze the events targeting those users to extract the most frequently invoked service models (i.e., <i>MostFrequent ServiceModelSets</i>, where <i>FSM</i> which means <i>Frequent ServiceModel</i> is an element of <i>MostFrequent ServiceModelSets</i>). In this final step, we use the FPGrowth algorithm [35], which is a classic Frequent Item Mining and Association Rules Mining approach known for its efficiency and effectiveness when compared to other known algorithms like Apriori and Eclat [36]. The general process is presented in algorithm 1.</p>	<p>Целью здесь является информирование экспертов и разработчиков предметной области о необходимости потенциальных изменений в предопределенных моделях услуг. Более того, мы стремимся предоставить им потенциальные элементы, которые будут включены в это изменение. Для этого мы анализируем следы выполнения (т.е. события), которые были записаны, чтобы обнаружить новые шаблоны в данных. Сначала мы отфильтровываем трассировки для извлечения задач, которые не принадлежат какой-либо предопределенной модели сервиса (то есть, уникальные атомарные задачи, где <i>UAT</i>, что означает уникальную атомарную задачу, является элементом уникальных атомарных задач). Между тем, мы группируем атомарные события вместе по задаче (то есть, Группированные атомарные события, где <i>GAE</i>, что означает Группированное атомарное событие, является элементом Группированных атомарных событий). Во-вторых, мы извлекаем набор пользователей, которым нужны задачи вызванных сервисов (т. Е. Пользователи, где <i>U</i>, что означает «Пользователь», является элементом «Пользователи»). В-третьих, мы анализируем события, нацеленные на этих пользователей, чтобы извлечь наиболее часто используемые модели сервисов (т. Е. Наборы моделей сервисов Most Frequent, где <i>FSM</i>, что означает Frequent ServiceModel, является элементом наборов моделей сервисов MostFrequent). На этом последнем шаге мы используем алгоритм FP Growth [35], который является классическим подходом извлечения и ассоциации правил для частых предметов, известным своей эффективностью и результативностью по сравнению с другими</p>

	известными алгоритмами, такими как Apriori и Eclat [36]. Общий процесс представлен в алгоритме 1.
<p>The proposed algorithm takes as input two parameters. First, a log file containing a set of <i>Events</i>. Each <i>Event</i> in the log file is a tuple (<i>User</i>, <i>Task</i>, <i>ServiceModel</i>) where each tuple (i.e., event) represents a performed task, its targeted user and the service model it belongs to if any. Second, a value <math>P</math> with <math>0 &lt; P \leq 100</math> represents a percentage threshold after which the recommendation becomes relevant to the domain experts and developers. Note, this threshold affects only atomic tasks and is used as a minimum support to check whether an atomic task is frequently associated with a service model. As output, the algorithm returns a set of recommendations <math>R</math>. Each <math>r \in R</math> is a tuple (<math>SM</math>, <math>AT</math>) where <math>SM</math> is a predefined service model and <math>AT</math> is an atomic task than be used to extend <math>SM</math>.</p>	<p>Предложенный алгоритм принимает в качестве входных данных два параметра. Во-первых, файл журнала, содержащий набор событий. Каждое событие в файле журнала является кортежем (пользователь, задача, модель сервиса), где каждый кортеж (то есть событие) представляет выполненную задачу, ее целевого пользователя и модель сервиса, к которой он относится, если таковой имеется. Во-вторых, значение <math>P</math> с <math>0 &lt; P \leq 100</math> представляет собой процентный порог, после которого рекомендация становится актуальной для экспертов предметной области и разработчиков. Обратите внимание, что этот порог влияет только на элементарные задачи и используется в качестве минимальной поддержки для проверки того, часто ли элементарная задача связана с моделью сервиса. В качестве выходных данных алгоритм возвращает набор рекомендаций <math>R</math>. Каждый <math>r \in R</math> является кортежем (<math>SM</math>, <math>AT</math>), где <math>SM</math> - предопределенная модель обслуживания, а <math>AT</math> - элементарная задача, которую следует использовать для расширения <math>SM</math>.</p>
<p>Filtering the data allows to only extract the itemsets that are associated with the atomic tasks. However, the domain experts and the developers still have to configure the frequent itemset mining algorithm by setting the minimum support. The setup of the algorithm's parameters is needed to make the decision of adding the atomic task to the predefined service models.</p>	<p>Фильтрация данных позволяет извлекать только те наборы элементов, которые связаны с атомарными задачами. Тем не менее, экспертам в предметной области и разработчикам по-прежнему приходится настраивать алгоритм анализа наборов частых элементов, устанавливая минимальную поддержку. Настройка параметров алгоритма необходима для принятия решения о добавлении атомарной задачи в предопределенные модели обслуживания.</p>
<b>5 Experiments and results</b>	<b>5 Эксперименты и результаты</b>
<p>This section describes the experiments that we conducted on the recommendation layer. We start by describing the data preparation process. Then, we present the results of our experiments, before starting a discussion to interpret the results.</p>	<p>В этом разделе описываются эксперименты, которые мы провели на уровне рекомендаций. Начнем с описания процесса подготовки данных. Затем мы представляем результаты наших экспериментов, прежде чем начать обсуждение, чтобы интерпретировать результаты.</p>
<b>5.1 Data preparation</b>	<b>5.1. Подготовка данных</b>

To conduct our experiments successfully, we collected the data generated from each layer of our approach. At the bottom layer (i.e., context collection), the data consist of the users list and the values of their corresponding context factors. The directly above layer (i.e., context processing) generates the situations based on the values of context factors. The service management layer generates the data related to the service models. Service invocation generates the data about the services that are invoked by the users. We extracted the data using a developed simulation of the case study presented in Sect. 3 and extracted the QoS values from the WSDREAM dataset 2 collected by [7]. The dataset describes the realworld QoS evaluation results from 339 users on 5825 Web services. The dataset also contains the location information (e.g., country, autonomous system, latitude, longitude) of the users and the services. For more details, users can refer to the work of [7]. In our prototype, we map the country to the environment and the AS (i.e., autonomous system) to the infrastructure in both the services and the users files. We aggregated and preprocessed the data to generate three separate files. The files contain the following elements: – The *users* data file containing the information about the users (i.e., the values of their context factors and their situation) – The *services* data file containing the information about the services (e.g., the tasks they perform, location, etc.) – The *invocations* data file containing the information about the invoked services by each user (i.e., QoS parameters). To simplify the process of reconstructing the service models and filtering the atomic tasks, we also stored the preceding task when performing the current task and the service model it belongs to. Table 4 shows an extract of the invocations file, where the user column represents the identifiers of each user, the service column represents the identifiers of the invoked services, the SM column represents the service model from which the service was executed (indicating the user’s situation) and response time and throughput represent objective QoS data about the invocation of each service by each user. The *invocation* data file is available on csv format in the linked repository at footnote.3

Для успешного проведения наших экспериментов мы собрали данные, полученные на каждом уровне нашего подхода. На нижнем уровне (то есть, сбор контекста) данные состоят из списка пользователей и значений соответствующих им контекстных факторов. Прямо над уровнем (то есть обработка контекста) генерирует ситуации на основе значений факторов контекста. Уровень управления услугами генерирует данные, относящиеся к моделям услуг. Сервисный вызов генерирует данные о сервисах, которые вызываются пользователями. Мы извлекли данные, используя разработанное моделирование тематического исследования, представленного в разд. 3 и извлекли значения QoS из набора данных WSDREAM 2, собранного в [7]. Набор данных описывает результаты оценки реального мира QoS от 339 пользователей на 5825 веб-сервисах. Набор данных также содержит информацию о местоположении (например, страну, автономную систему, широту, долготу) пользователей и услуг. Более подробно пользователи могут обратиться к работе [7]. В нашем прототипе мы сопоставляем страну с окружающей средой, а AS (то есть автономную систему) с инфраструктурой как в файлах служб, так и в файлах пользователей. Мы агрегировали и предварительно обрабатывали данные для создания трех отдельных файлов. Файлы содержат следующие элементы: - файл данных пользователей, содержащий информацию о пользователях (т.е. значения их контекстных факторов и их ситуации); - файл данных служб, содержащий информацию об услугах (например, задачи, которые они выполняют, местоположение и т. д.) - файл данных о вызовах, содержащий информацию о вызываемых услугах для каждого пользователя (т. е. параметры QoS). Чтобы упростить процесс восстановления моделей сервисов и фильтрации элементарных задач, мы также сохранили предыдущую задачу при выполнении текущей задачи и модель сервиса, к которой она относится. В таблице 4 показан фрагмент файла вызовов, где столбец пользователя представляет идентификаторы каждого пользователя,

	<p>столбец службы представляет идентификаторы вызванных служб, столбец SM представляет модель службы, из которой была выполнена служба (с указанием ситуации пользователя). ), а время отклика и пропускная способность представляют объективные данные QoS о вызове каждой услуги каждым пользователем. Файл данных о вызовах доступен в формате csv в связанном репозитории на сноске. 3</p>
<p><b>Algorithm 1:</b> Algorithm for Service Model Recommendation for Developers and Domain Experts</p>	<p>Алгоритм 1: Алгоритм модели рекомендательного сервиса для разработчиков и экспертов в предметной области</p>



**Input** : The set of recorded events *Events*, such that each event is described as:  
*Event* = (*User*, *Task*, *ServiceModel*) and *P* being the threshold after which the service is considered relevant for recommendation

**Output**: Set of pairs *R* = (*SM*, *AT*), Where *SM* is the predefined Service Model and *AT* is the recommended Atomic Task for integration

```

1 R ← ∅
2 AtomicEvents ← getAtomicEvents(Events)
   // Events with empty ServiceModel field
3 GroupedAtomicEvents ← AtomicEvents.GroupbyTask()
4 UniqueAtomicTasks ←
   AtomicEvents.GetUniqueTasks()
5 foreach UAT ∈ UniqueAtomicTasks do
6   foreach GAE in GroupedAtomicEvents.getGroup(UAT) do
7     Users ← GAE.getUsers()
8     NumberOfUsers ← Users.getCount()
9     ServiceModels ← ∅
10    foreach U ∈ Users do
11      | ServiceModels.Add(U.getServiceModels())
12    end
13    FrequentServiceModels ←
      FPGrowth(ServiceModels, NumberOfUsers ×
         $\frac{P}{100}$ )
14    if FrequentServiceModels ≠ ∅ then
15      | MostFrequentServiceModelSets ←
        | FrequentServiceModels.pop(3)
        | // get first three results
16      | foreach
        | FSM ∈ MostFrequentServiceModelSets do
17        | | R.Add(FSM, UAT)
18      | end
19    end
20  end
21 end
22 return R

```

<b>5.2 Results</b>	5.2. Результаты
<b>5.2.1 QoS-aware service recommendation for end users</b>	
<b>Experiment Setup</b> We ran an implementation of the userbased AutoRec (U-AutoRec) approach on the data collected in the <i>invocations</i> file using TensorFlow.4 The experiment was conducted using a laptop with a seventh generation Intel i5 processor, 8GB of RAM and 6MB of cache memory. The model parameters chosen throughout the experiments are presented in Table 5.	Настройка эксперимента Мы запустили реализацию пользовательского подхода AutoRec (U-AutoRec) для данных, собранных в файле вызовов с использованием TensorFlow.4. Эксперимент проводился на ноутбуке с процессором Intel i5 седьмого поколения, 8 ГБ ОЗУ и 6 МБ кэш-памяти. Параметры модели, выбранные в ходе экспериментов, представлены в таблице 5.
As mentioned in Sect. 4.6.1, the knowledge about QoS properties is not complete in real use cases, as that would imply that every user has invoked every available service. To reflect this aspect of sparsity on the WSDREAM dataset, we used less of one-third of the data to build the model. Particularly, we built the model using four different matrix density levels: <i>MatrixDensity</i> = 5%, 10%, 20%, 30%. <i>MatrixDensity</i> = 5% means that we randomly select 5% of the data to predict the remaining 95%. Each experiment was ran five times.	Как упомянуто в разд. 4.6.1 знание о свойствах QoS не является полным в реальных случаях использования, поскольку это будет означать, что каждый пользователь вызывал все доступные службы. Чтобы отразить этот аспект разреженности в наборе данных WSDREAM, мы использовали менее трети данных для построения модели. В частности, мы построили модель, используя четыре разных уровня плотности матрицы: <i>MatrixDensity</i> = 5%, 10%, 20%, 30%. Матрица плотности = 5% означает, что мы случайным образом выбираем 5% данных, чтобы предсказать оставшиеся 95%. Каждый эксперимент проводился пять раз.
<b>Evaluation Metrics</b> To evaluate our U-AutoRec-based model when compared to other QoS prediction methods, we focused on the accuracy of the predicted values. We define accuracy as a function describing the distance between the observed data and the predicted data. In our case, the observed data correspond to the QoS data recorded in the WSDREAM dataset, meaning response time and throughput data. The predicted data correspond to the data predicted by our model as an output. In this context, we use MAE and RMSE as two basic and popular error metrics that are commonly used in evaluating recommender systems [15]. While MAE measures the mean magnitude of errors over a set of prediction, RMSE is the square root of the average of squared differences between prediction and actual observation. This makes RMSE more interesting when large errors have a lot more negative effects on the	Метрики оценки Чтобы оценить нашу модель на основе U-AutoRec по сравнению с другими методами прогнозирования QoS, мы сосредоточились на точности прогнозируемых значений. Мы определяем точность как функцию, описывающую расстояние между наблюдаемыми данными и предсказанными данными. В нашем случае наблюдаемые данные соответствуют данным QoS, записанным в наборе данных WSDREAM, что означает время отклика и данные о пропускной способности. Предсказанные данные соответствуют данным, предсказанным нашей моделью в качестве выходных данных. В этом контексте мы используем MAE и RMSE как две основные и популярные метрики ошибок, которые обычно используются при оценке рекомендательных систем [15]. В то время как MAE измеряет среднюю величину ошибок в наборе

system than small errors. MAE and RMSE are defined as:	прогнозов, RMSE - это квадратный корень из среднего квадрата различий между прогнозированием и фактическим наблюдением. Это делает RMSE более интересным, когда большие ошибки оказывают гораздо большее негативное влияние на систему, чем маленькие ошибки. MAE и RMSE определяются как:																														
$MAE = \frac{\sum_{u,s}  r_{u,s} - \hat{r}_{u,s} }{N}$ $RMSE = \sqrt{\frac{\sum_{u,s} (r_{u,s} - \hat{r}_{u,s})^2}{N}}$																															
where $r_{u,s}$ denotes the expected QoS value of service $s$ observed by user $u$ , $\hat{r}_{u,s}$ is the predicted QoS value and $N$ is the number of values.	где $r_{u,s}$ обозначает ожидаемое значение QoS услуг, наблюдаемых пользователем $u$ , $\hat{r}_{u,s}$ - это прогнозируемое значение QoS, а $N$ - количество значений.																														
<b>Table 4</b> Example of the <i>invocation</i> data file generated by the SMARTROAD system	Таблица 4 Пример файла данных вызова, созданного системой SMARTROAD																														
<table><tr><th>User</th><th>SM</th><th>Service</th><th>Throughput</th><th>Response time</th></tr><tr><td>User_7</td><td></td><td>Weather_serv_03</td><td>24.1</td><td>5.12</td></tr><tr><td>User_2</td><td>Risky vehicle</td><td>SpeedAdaptation_serv_2</td><td>14.5</td><td>0.48</td></tr><tr><td>User_42</td><td>Risky vehicle</td><td>Garage_serv_4</td><td>5.4</td><td>1.45</td></tr><tr><td>User_18</td><td></td><td>Towing_serv_06</td><td>16.34</td><td>2.15</td></tr><tr><td>User_68</td><td></td><td>Traffic_serv_01</td><td>45.14</td><td>1.84</td></tr></table>	User	SM	Service	Throughput	Response time	User_7		Weather_serv_03	24.1	5.12	User_2	Risky vehicle	SpeedAdaptation_serv_2	14.5	0.48	User_42	Risky vehicle	Garage_serv_4	5.4	1.45	User_18		Towing_serv_06	16.34	2.15	User_68		Traffic_serv_01	45.14	1.84	
User	SM	Service	Throughput	Response time																											
User_7		Weather_serv_03	24.1	5.12																											
User_2	Risky vehicle	SpeedAdaptation_serv_2	14.5	0.48																											
User_42	Risky vehicle	Garage_serv_4	5.4	1.45																											
User_18		Towing_serv_06	16.34	2.15																											
User_68		Traffic_serv_01	45.14	1.84																											
<b>Table 5</b> Model parameters	Таблица 5 Параметры модели																														
<table><tr><th>N° HL</th><th><math>k</math></th><th>Learning rate</th><th><math>\lambda</math></th><th>Batch size</th><th>Optimizer</th><th>A. Function</th></tr><tr><td>1</td><td>500</td><td>0.0001</td><td>0.1</td><td>32</td><td>Adam</td><td>Sigmoid</td></tr></table>	N° HL	$k$	Learning rate	$\lambda$	Batch size	Optimizer	A. Function	1	500	0.0001	0.1	32	Adam	Sigmoid																	
N° HL	$k$	Learning rate	$\lambda$	Batch size	Optimizer	A. Function																									
1	500	0.0001	0.1	32	Adam	Sigmoid																									
<b>Performance Comparison</b> We compare the performance of U-AutoRec with the performance of several other methods. In the literature, each method was evaluated on a different set of <i>MatrixDensity</i> values. This makes the comparison highly challenging as source code is rarely publicly available. Hence, in this paper, we compared the U-AutoRec method with the methods reported in the WS-DREAM package [37]. The package reports the performance of several methods. Some of these methods are based on neighborhood [38–	Сравнение производительности Мы сравниваем производительность U-AutoRec с производительностью нескольких других методов. В литературе каждый метод оценивался по разному набору значений MatrixDensity. Это делает сравнение очень сложным, так как исходный код редко общедоступен. Следовательно, в этой статье мы сравнили метод U-AutoRec с методами, описанными в пакете WS-DREAM [37]. Пакет сообщает о производительности																														

<p>40], some on models [41–43] and others are location-aware [44– 48]. Table 6 shows the mean MAE and RMSE values of different prediction methods in response time (RT) measured in seconds and throughput (TP) measured in Kbps, using 5, 10, 20 and 30 percent as <i>MatrixDensity</i> (MD) values.</p>	<p>нескольких методов. Некоторые из этих методов основаны на окрестности [38–40], некоторые на моделях [41–43], а другие учитывают местоположение [44–48]. В таблице 6 приведены средние значения MAE и RMSE различных методов прогнозирования во времени отклика (RT), измеренные в секундах, и пропускной способности (TP), измеренные в кбит / с, с использованием 5, 10, 20 и 30 процентов в качестве значений <i>MatrixDensity</i> (MD).</p>
<p><b>5.2.2 Service model recommendation for developers and domain experts</b></p>	<p>5.2.2 Модель рекомендательного сервиса для разработчиков и экспертов в предметной области</p>
<p>To validate our approach, we injected records of executed atomic services in the invocations file with different probabilities. The objective is to prove that the proposed algorithm is capable of extracting and recommending relevant atomic tasks for them to integrate into existing service models, thus achieving continuous improvements. Hence, the accuracy metric for this part is the capacity of our algorithm to discover these injected patterns. We considered two situations where we injected the following patterns.</p>	<p>Чтобы проверить наш подход, мы внедрили записи выполненных атомарных сервисов в файл вызовов с различными вероятностями. Цель состоит в том, чтобы доказать, что предлагаемый алгоритм способен извлекать и рекомендовать соответствующие элементарные задачи для их интеграции в существующие модели обслуживания, что позволяет достичь непрерывных улучшений. Следовательно, метрика точности для этой части - это способность нашего алгоритма обнаруживать эти введенные шаблоны. Мы рассмотрели две ситуации, когда мы вводили следующие шаблоны.</p>
<p>– That 80% of the users who encountered a Risky Vehicle State situation have searched for a Gas Station to fill the gas tank of their vehicle (i.e., GSSS). Note, the GSSS service is not part of any predefined service model (see Fig. 2). – That 40% of the users who encountered a Risky Vehicle State situation have searched for a Towing service to transport their vehicles (i.e., TSS).</p>	<p>- 80% пользователей, которые столкнулись с ситуацией с опасным состоянием автомобиля, искали заправочную станцию, чтобы заполнить бензобак своего автомобиля (то есть GSSS). Обратите внимание, что услуга GSSS не является частью какой-либо заранее определенной модели обслуживания (см. Рис. 2). - 40% пользователей, которые столкнулись с опасной ситуацией с состоянием автомобиля, искали услугу буксировки для перевозки своих транспортных средств (т. Е. TSS).</p>
<p>We ran an implementation of algorithm 1 in Python, which is available in the linked repository at footnote .5 As inputs, we entered the path to the <i>invocation</i> file and <math>P = 50</math> is the threshold percentage of the users who have performed the task before it is linked to the service model. While no results were found for the TSS service, the GSSS was recommended to be integrated in the RiskyVehicleState as expected. We</p>	<p>Мы запустили реализацию алгоритма 1 в Python, который доступен в связанном репозитории в сноске .5 В качестве входных данных мы ввели путь к файлу вызова, а <math>P = 50</math> - это пороговый процент пользователей, которые выполнили задачу перед ней. связан с моделью сервиса. Пока не было найдено никаких результатов для сервиса TSS, GSSS было рекомендовано интегрировать в Risky</p>

expect the algorithm to reveal interesting recommendations when applied on real data.	Vehicle State, как и ожидалось. Мы ожидаем, что алгоритм покажет интересные рекомендации применительно к реальным данным.
<b>5.3 Analysis and discussion</b>	<b>5.3. Анализ и обсуждение</b>
The results presented in Table 6 show that model-based methods generally outperform those based on location awareness or neighborhood. Furthermore, U-AutoRec performs exceptionally well even when compared to other model-based methods. This is especially true, when analyzing the performance of the approach on throughput (TP) data, where U-AutoRec achieves 33.2% gains in MAE and 35.3% better in RMSE, when compared with the best approaches over all matrix density values individually. On the throughput (TP) data, U-AutoRec also scores an improvement of 32.1% in MAE and a 29.4% gain on RMSE.	Результаты, представленные в Таблице 6, показывают, что основанные на модели методы обычно превосходят методы, основанные на осведомленности о местоположении или соседстве. Кроме того, U-AutoRec работает исключительно хорошо, даже по сравнению с другими методами на основе моделей. Это особенно верно при анализе производительности подхода на основе данных о пропускной способности (TP), когда U-AutoRec достигает 33,2% прироста в MAE и 35,3% лучше в RMSE по сравнению с лучшими подходами по всем значениям плотности матрицы в отдельности. По данным о пропускной способности (TP), U-AutoRec также получил улучшение на 32,1% в MAE и увеличение на 29,4% на RMSE.
While U-AutoRec already outperforms the other methods in most cases, there is still room for improvement. Indeed, with the huge number of combination that can be used to finetune the model (i.e., number of iterations, optimizers, number of hidden nodes, activation functions, etc.), it is most likely that our experiment did not reach the limits of the AutoRec framework. Extensive experiments and finetuning techniques are needed in this direction to identify the best parameters for the model.	Хотя U-AutoRec уже превосходит другие методы в большинстве случаев, все еще есть возможности для улучшения. Действительно, с огромным количеством комбинаций, которые можно использовать для тонкой настройки модели (т. Е. Числом итераций, оптимизаторов, количества скрытых узлов, функций активации и т. Д.), наиболее вероятно, что наш эксперимент не достиг пределов платформы AutoRec. В этом направлении необходимы обширные эксперименты и методы тонкой настройки, чтобы определить наилучшие параметры для модели.
We have also ran experiments using I-AutoRec, which follows the same model as U-AutoRec with the difference that it relies on the items representations rather than the users representations to build the model. I-AutoRec revealed some interesting results. I-AutoRec performs generally better on throughput data, reaching an MAE = 14.63 and an RMSE = 38.26, when MD = 10%, and an MAE = 12.55 and an RMSE = 34.77, when MD = 20%. However, U-AutoRec still outperforms I-AutoRec in both response time (RT) and throughput (TP) data. This leads us to believe that response time data are more sensitive to the	Мы также провели эксперименты с использованием I-AutoRec, который следует той же модели, что и U-AutoRec, с той разницей, что для построения модели используются представления элементов, а не представления пользователей. I-AutoRec показал несколько интересных результатов. I-AutoRec обычно работает лучше с данными о пропускной способности, достигая MAE = 14,63 и RMSE = 38,26, когда MD = 10%, и MAE = 12,55 и RMSE = 34,77, когда MD = 20%. Тем не менее, U-AutoRec по-прежнему превосходит I-AutoRec как по времени отклика (RT), так и по данным о



<p>features of the users, while throughput data are also sensitive to the features of the services. This is logical as response time would change depending on the network state of the users (e.g., signal strength), while throughput depends on the network state of the service's supporting infrastructure and that of the user.</p>	<p>пропускной способности (TP). Это заставляет нас полагать, что данные времени отклика более чувствительны к функциям пользователей, а данные о пропускной способности также чувствительны к функциям сервисов. Это логично, поскольку время отклика будет изменяться в зависимости от состояния сети пользователей (например, уровня сигнала), тогда как пропускная способность зависит от состояния сети поддерживающей инфраструктуры службы и пользователя.</p>
<p>Table 6 Performance comparison between the QoS prediction models</p>	<p>Таблица 6 Сравнение характеристик между моделями прогнозирования QoS</p>

**Table 6** Performance comparison between the QoS prediction models

Methods	MD = 5%						MD = 10%						MD = 20%						MD = 30%					
	RT			TP			RT			TP			RT			TP			RT			TP		
	MAE	RMSE		MAE	RMSE		MAE	RMSE		MAE	RMSE		MAE	RMSE		MAE	RMSE		MAE	RMSE		MAE	RMSE	
UIPCC [38]	0.6253	1.3879		26.7568	60.7985		0.5815	1.3302		22.3700	54.4563		0.4498	1.1968		18.9276	48.2950		0.4110	1.1422		17.0797	45.0599	
ADF [39]	0.6094	1.3613		24.9961	60.7939		0.5443	1.2924		21.5013	54.2893		0.4636	1.1898		16.6536	45.2008		0.4276	1.1398		14.8244	41.7186	
NRCF [40]	0.5532	1.4547		23.3275	59.9498		0.4905	1.3678		18.8571	52.9977		0.4261	1.2581		14.3444	44.5142		0.4059	1.1975		12.8267	40.7493	
RegionKNN [44]	0.5883	1.5426		25.6324	67.8678		0.5477	1.5129		24.8380	67.5510		0.5158	1.5214		24.0361	66.1760		0.5091	1.5193		23.7984	65.5971	
LACF [45]	0.6374	1.4436		23.1685	58.9666		0.5659	1.3420		19.6257	53.1060		0.4827	1.2258		16.6669	47.6247		0.4453	1.1720		15.2358	44.7729	
LBR [46]	0.5499	1.4741		18.3187	56.0215		0.4802	1.2858		15.4272	47.0197		0.4300	1.1610		13.6512	41.3552		0.4103	1.1214		12.9824	39.2269	
HMF [48]	0.5595	1.5248		19.1320	58.7088		0.4815	1.3105		15.7187	48.3461		0.4266	1.1661		13.6319	41.6678		0.4072	1.1178		12.7767	39.0613	
LoRec [47]	0.6479	1.3957		27.7773	62.5533		0.5557	1.3087		24.7118	58.4275		0.4659	1.2000		21.7440	54.0139		0.4437	1.1504		20.5338	52.0747	
PMF [41,49]	0.5690	1.5371		19.0946	57.9078		0.4866	1.3163		15.9741	48.0393		0.4308	1.1690		13.9085	41.7277		0.4082	1.1206		13.1782	39.4635	
NMF [42,50]	0.5456	1.4727		18.8824	57.5177		0.4775	1.2824		15.5717	47.7903		0.4291	1.1616		13.5259	41.6482		0.4063	1.1191		12.7851	39.5068	
BiasedMF [43]	0.5934	1.3821		21.8369	56.8575		0.5135	1.2625		17.8522	48.3275		0.4569	1.1786		14.9224	42.1408		0.4280	1.1416		13.6980	39.6656	
LN-LFM [43]	0.5602	1.3065		20.4018	52.4209		0.5007	1.2284		17.7273	46.9713		0.4512	1.1541		16.3428	44.1434		0.4351	1.1282		16.0625	43.4570	
Our model	0.361	1.010		13.542	37.119		0.3343	0.9035		10.360	30.668		0.2953	0.8019		8.644	26.569		0.2685	0.7688		7.952	24.562	

<p><b>6 Related works</b></p> <p>Service-based systems and their application have evolved a lot over the last two decades. Recently, researchers have focused on using the advances in AI planning to automate the service composition process [51]. One of the most popular techniques are those based on HTN (Hierarchical Task Network). Among these, figures the work of Wu et al. [52], where the authors propose a HTN-based (Hierarchical Task Network) planner called SHOP2 to automatically compose service in DAML-S. Other techniques like [53] rely on the advances in Semantic Web, using ontologies to automatically compose services. However, both approaches are still incapable of handling naturally complex real- world applications. Which is why most service solutions in industry still use manual or semiautomatic techniques in which the expertise of domain experts and developers is strongly needed [9].</p>	<p><b>6 Сопутствующие работы</b></p> <p>За последние два десятилетия сервисные системы и их применение сильно изменились. В последнее время исследователи сосредоточились на использовании достижений в области планирования ИИ для автоматизации процесса формирования услуг [51]. Одним из наиболее популярных методов являются методы, основанные на HTN (иерархическая сеть задач). Среди них фигурируют работы Wu et al. [52], где авторы предлагают планировщик на основе HTN (Hierarchical Task Network) под названием SHOP2 для автоматического создания сервиса в DAML-S. Другие методы, такие как [53], опираются на достижения в семантической сети, используя онтологии для автоматического создания сервисов. Однако оба подхода по-прежнему не способны обрабатывать естественные сложные приложения реального мира. Вот почему в большинстве сервисных решений в промышленности все еще используются ручные или полуавтоматические методы, в которых настоятельно необходимы знания экспертов и разработчиков в области [9].</p>
<p>However, manual and semiautomatic techniques are not without their challenges. While domain experts usually have the necessary knowledge to compose relevant service models, they cannot create personalized services that are tailored to the preferences of the users. This lack in personalization aspects was discussed by [9]. In fact, in their survey, only 8% of the included works have investigated personalization. This problem gains more emphasis when the environments are of a distributed and ubiquitous nature (e.g., smart environments). To tackle this problem, context-aware approaches have been proposed by many researchers to integrate context information at different steps of the composition process [26,54]. Most of the proposed approaches rely on ontologies to represent context information and their innate use of first-order logic to reason on the collected information. In our approach, we integrated contextual information to extract the situation of the user through a metamodeling approach for more generality.</p>	<p>Тем не менее, ручные и полуавтоматические методы не обходятся без своих проблем. Хотя специалисты в области обычно имеют необходимые знания для составления соответствующих моделей услуг, они не могут создавать персонализированные услуги, которые адаптированы к предпочтениям пользователей. Этот недостаток в аспектах персонализации обсуждался в [9]. Фактически, в их обзоре только 8% включенных работ исследовали персонализацию. Эта проблема становится все более актуальной, когда среды имеют распределенный и повсеместный характер (например, интеллектуальные среды). Для решения этой проблемы многие исследователи предложили контекстно-зависимые подходы для интеграции контекстной информации на разных этапах процесса компоновки [26,54]. Большинство предлагаемых подходов основаны на онтологиях для представления контекстной информации и их врожденного использования логики первого</p>

	<p>порядка для обоснования собранной информации. В нашем подходе мы интегрировали контекстную информацию, чтобы извлечь ситуацию пользователя через метамодельный подход для большей общности.</p>
<p>Even though these methods are capable of identifying the particular needs, requirements and preferences of the users, domain experts and developers still find themselves unable to capture the different deviations and variations at user level and enact them in the defined service models. To this end, it was necessary to introduce new techniques to support them in this task. Existing methods suggest analyzing existing service models to help the users (i.e., domain expert and developers) in the creation of new service models [55–57]. In our approach, we analyze the history of used services to recommend services to be integrated into existing service models or to be added as new variability models.</p>	<p>Несмотря на то, что эти методы способны идентифицировать конкретные потребности, требования и предпочтения пользователей, эксперты в области и разработчики по-прежнему не могут уловить различные отклонения и различия на уровне пользователя и применить их в определенных моделях услуг. С этой целью было необходимо ввести новые технологии для поддержки их в этой задаче. Существующие методы предлагают анализ существующих моделей услуг, чтобы помочь пользователям (т. Е. Эксперту в области и разработчикам) в создании новых моделей услуг [55–57]. В нашем подходе мы анализируем историю использованных услуг, чтобы рекомендовать услуги для интеграции в существующие модели услуг или для добавления в качестве новых моделей изменчивости.</p>
<p>Service selection is another important problem when dealing with distributed and ubiquitous environments, given the number of available services and the increasing mobility and dynamics of the users. Recently, recommender systems have been widely investigated, to great success, in dealing with this problem of selecting the best services for a particular user or a set of users [7]. Collaborative filtering techniques in particular have been very successful in dealing with this problem. Most of the methods proposed in the literature use registered QoS properties to recommend the best services. These methods can usually be classified into three categories: (1) neighborhood-based methods, relying on the similarity between the users in terms of past recorded QoS parameters to predict the values of missing values of QoS properties [38– 40]; (2) location-aware methods, using the similarity between the user profiles in general and location in particular to predict the values of missing values of QoS properties [44–47]; and (3) model-based methods, using machine learning techniques to build models that are capable of predicting the</p>	<p>Выбор услуг является еще одной важной проблемой при работе с распределенными и вездесущими средами, учитывая количество доступных услуг и растущую мобильность, и динамику пользователей. В последнее время рекомендательные системы были широко исследованы, с большим успехом, в решении этой проблемы выбора лучших услуг для конкретного пользователя или группы пользователей [7]. В частности, методы совместной фильтрации были очень успешными в решении этой проблемы. Большинство методов, предложенных в литературе, используют зарегистрированные свойства QoS, чтобы рекомендовать лучшие услуги. Эти методы обычно можно разделить на три категории: (1) основанные на соседстве методы, основанные на сходстве между пользователями в терминах прошлых зарегистрированных параметров QoS для прогнозирования значений отсутствующих значений свойств QoS [38–40]; (2) методы с учетом местоположения, использующие сходство между профилями</p>

<p>values of missing values of QoS properties [41–43]. Model-based methods generally perform better than the other categories. In this paper, we build a model using the AutoRec framework [33], which outperforms other model-based methods.</p>	<p>пользователей в целом и местоположением в частности для прогнозирования значений пропущенных значений свойств QoS [44–47]; и (3) основанные на модели методы, использующие методы машинного обучения для построения моделей, способных прогнозировать значения пропущенных значений свойств QoS [41–43]. Методы на основе моделей обычно работают лучше, чем другие категории. В этой статье мы строим модель, используя платформу AutoRec [33], которая превосходит другие основанные на модели методы.</p>
<p><b>7 Conclusion and future work</b></p>	<p><b>7 Заключение и будущая работа</b></p>
<p>Nowadays, ubiquitous environments have made it difficult to propose relevant services to the users. First, due to the increasing mobility of the users and the different devices connected to the Internet, capturing the situation of the users is challenging. Second, the proliferation of available services in these environments makes the task of selecting the best services for each user problematic. Third, the variety of user requirements and needs makes the task of designing service models that are capable of satisfying each user needs highly complex.</p>	<p>В настоящее время повсеместно распространенные среды затрудняют предоставление пользователям соответствующих услуг. Во-первых, из-за растущей мобильности пользователей и различных устройств, подключенных к Интернету, уловить ситуацию пользователей сложно. Во-вторых, распространение доступных услуг в этих средах делает задачу выбора лучших услуг для каждого пользователя проблематичной. В-третьих, разнообразие требований и потребностей пользователей делает задачу разработки моделей услуг, способных удовлетворить потребности каждого пользователя, весьма сложной.</p>
<p>In this paper, we proposed a recommendation-based system for service composition targeting smart environments. The main contributions of this paper can be summarized in four points: (1) a conceptual architecture for a context-aware recommendation-based system for service composition; (2) a metamodel to represent the situation of the users through context information; (3) a model based on the AutoRec framework for the prediction of QoS values and (4) an algorithm for recommending tasks to be integrated in existing service models or creating new variability models. The conducted experiments show the efficiency and effectiveness of the recommendation policies proposed. QoS predictionwise, we achieved an improvement of 30% on MAE over existing methods. Task recommendation-wise, the proposed algorithm produces the expected results when applied to synthetic data.</p>	<p>В этой статье мы предложили основанную на рекомендациях систему составления услуг, ориентированную на интеллектуальные среды. Основные вклады этого документа могут быть обобщены в четырех пунктах: (1) концептуальная архитектура для контекстно-ориентированной системы рекомендаций на основе состава услуг; (2) метамодель для представления ситуации пользователей через контекстную информацию; (3) модель, основанная на структуре AutoRec для прогнозирования значений QoS, и (4) алгоритм для рекомендации задач для интеграции в существующие модели обслуживания или создания новых моделей изменчивости. Проведенные эксперименты показывают эффективность и действенность предложенных рекомендаций. Согласно прогнозу QoS, мы достигли 30-процентного улучшения MAE по сравнению с</p>



	существующими методами. По рекомендации, предлагаемый алгоритм дает ожидаемые результаты при применении к синтетическим данным.
Deep learning frameworks have shown their efficiency and effectiveness on many application domains. The application of deep learning algorithms in recommender systems is supposed to enhance their performance even further [58]. Research has shown the improvements in performance when applied to standard datasets such as MovieLens. Hence, it would be beneficial to explore the use of such frameworks to build new models for QoS prediction for service recommendation, as they can potentially provide better performances.	Среды глубокого обучения показали свою эффективность и действенность во многих областях применения. Предполагается, что применение алгоритмов глубокого обучения в рекомендательных системах еще более повысит их производительность [58]. Исследования показали улучшения в производительности при применении к стандартным наборам данных, таким как MovieLens. Следовательно, было бы полезно изучить использование таких структур для создания новых моделей для прогнозирования QoS для рекомендаций по обслуживанию, поскольку они потенциально могут обеспечить более высокую производительность.
To date, there exists no dataset or log files recording the behavior of composite services. The service computing community could highly benefit from such data. The data could be used to explore the effect of the composition on the selection algorithm, meaning that considering two services $s_1$ and $s_2$ offering the same task $t_1$ , where $s_1$ is better than $s_2$ and two other services $s_3$ and $s_4$ offering the same task $t_2$ , where $s_3$ is better than $s_4$ ; would the composition of $s_1 \rightarrow s_3$ be better than the composition of $s_2 \rightarrow s_3$ ? and if not, then the research should maybe focus on predicting QoS over service patterns rather than individual services.	На сегодняшний день не существует ни набора данных, ни файлов журналов, фиксирующих поведение составных сервисов. Сообщество сервисных вычислений может очень выиграть от таких данных. Эти данные можно использовать для изучения влияния композиции на алгоритм выбора. Это означает, что с учетом двух сервисов $s_1$ и $s_2$ , предлагающих одну и ту же задачу $t_1$ , где $s_1$ лучше, чем $s_2$ , и двух других сервисов $s_3$ и $s_4$ , предлагающих одну и ту же задачу $t_2$ , где $s_3$ лучше, чем $s_4$ ; будет состав $s_1 \rightarrow s_3$ лучше, чем состав $s_2 \rightarrow s_3$ ? и если нет, то исследование должно быть сосредоточено на прогнозировании QoS по шаблонам обслуживания, а не по отдельным услугам.
Another research direction that is worth investigating in recommender systems in general, and in service recommendation in particular, is the impact of these systems on the performance of the recommended services. Indeed, recommending the same service to a myriad of users could potentially be dangerous, as the service could be overloaded with multiple requests at the same time, which we assume could result in a serious drop in its QoS. Having the knowledge about the computing environment of the services could help investigate how many requests	Другое направление исследований, которое стоит исследовать в рекомендательных системах в целом и в рекомендациях по обслуживанию в частности, - это влияние этих систем на производительность рекомендуемых услуг. Действительно, рекомендация одной и той же службы множеству пользователей потенциально может быть опасной, поскольку служба может быть перегружена несколькими запросами одновременно, что, как мы полагаем, может привести к серьезному снижению QoS. Знание

<p>can a service handle before it suffers QoS drops. New recommender systems would need to take this information into account in the selection process to avoid negative effects on the service.</p>	<p>вычислительной среды сервисов может помочь выяснить, сколько запросов может обработать сервис, прежде чем он потеряет QoS. Новые рекомендательные системы должны будут учитывать эту информацию в процессе выбора, чтобы избежать негативного воздействия на услугу.</p>
<p><b>Acknowledgements</b> This project was financially supported by CAMPUS FRANCE (PHC TOUBKAL 2017 (French-Morocco bilateral program) Grant Number: 36804YH).</p>	
<p><b>References</b> 1. Kondepudi SN, Ramanarayanan V, Jain A, Singh GN, Nitin Agarwal NK, Kumar R, Singh R, Bergmark P, Hashitani T, Gemma P et al (2014) Smart sustainable cities analysis of definitions. The ITU-T focus group for smart sustainable cities 2. DESA United Nations (2015) World urbanization prospects: the 2014 revision. United Nations Department of Economics and Social Affairs, Population Division, New York, NY, USA 3. SchaffersH,Komninou N, Pallot M, Trousse B, NilssonM,Oliveira A (2011) Smart cities and the future internet: towards cooperation frameworks for open innovation. In: The future internet. Springer, Berlin, pp 431–446 4. Bieberstein N, Bose S, Walker L, Lynch A (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. IBM Syst J 44(4):691–708 5. Baresi L, Di Nitto E, Ghezzi C, Guinea S (2007) A framework for the deployment of adaptable web service compositions. Serv Oriented Comput Appl 1(1):75–91 6. Hashem IAT, Chang V, Anuar NB, Ibrar Yaqoob KA, Gani A, Ahmed E, Chiroma H (2016) The role of big data in smart city. Int J Inf Manag 36(5):748–758 7. Zheng Z, Zhang Y, LyuMR (2014) Investigating qos of real-world web services. IEEE Trans Serv Comput 7(1):32–39 8. JinX, Chun S, Jung J, LeeK-H (2017)Afast and scalable approach for iot service selection based on a physical service model. Inf Syst Front 19(6):1357–1372 9. Sheng QZ, Qiao X, Vasilakos AV, Szabo C, Bourne S, Xiaofei X (2014) Web services composition: a decade’s overview. Inf Sci 280:218–238 10. Casati F, Shan M-C (2001) Dynamic and adaptive composition of e-services. Inf Syst 26(3):143–163 11. Fang J, Hu S, Han Y (2004) A</p>	

service interoperability assessment model for service composition. In: IEEE international conference on services computing, 2004 (SCC 2004). Proceedings. 2004, pp 153–158

12. Konstan JA, Riedl J (2012) Recommender systems: from algorithms to user experience. *User Model User Adapt Interact* 22(1):101–123

13. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: *The adaptive web*. Springer, pp 291–324

14. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM conference on computer supported cooperative work. CSCW '94*. ACM, New York, NY, USA, pp 175–186

15. Candillier L, Meyer F, Boullé M (2007) Comparing state-of-the-art collaborative filtering systems. In: Petra P (ed) *Machine learning and data mining in pattern recognition*. Springer, Berlin, pp 548–562

16. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37

17. Lee M, Choi P, Woo Y (2002) A hybrid recommender system combining collaborative filtering with neural network. In: *International conference on adaptive hypermedia and adaptive web-based systems*. Springer, pp 531–534

18. Bobadilla J, Ortega F, Hernando A, Alcalá J (2011) Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowl Based Syst* 24(8):1310–1316

19. Aggarwal CC, Wolf JL, Wu K-L, Yu PS (1999) Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In: *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '99*. ACM, New York, NY, USA, pp 201–212

20. Son J, Kim SB (2017) Content-based filtering for recommendation systems using multiattribute networks. *Expert Syst Appl* 89:404–412

21. Burke R (2002) Hybrid recommender systems: survey and experiments. *User Model User Adapt Interact* 12(4):331–370

22. Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggles P (1999) Towards a better understanding of context and contextawareness. In: *International symposium on handheld and ubiquitous computing*. Springer, pp 304–

307 23. Pinheiro MK, Souveyet C (2018) Supporting context on software applications: a survey on context engineering. *Modélisation et utilisation du contexte* 2(1):1–29. <https://doi.org/10.21494/ISTE.OP.2018.0275> 24. WorldHealth Organization et al (2018) Global status report on road safety 2018. World Health Organization, Geneva 25. National Committee for the Prevention of Traffic Accidents (Morocco). Provisional assessment of road accidents for 2017, 2016 26. Chang J, YaoW, Li X (2017) The design of a context-aware service system in intelligent transportation system. *Int J Distrib Sens Netw* 13(10):1550147717738165 27. Ait-Cheik-BihiW, Nait-Sidi-Moh A, Bakhouya M, Gaber J, Wack M (2012) Transportml platform for collaborative location-based services. *Serv Oriented Comput Appl* 6(4):363–378 28. Dijkstra EW (1982) On the role of scientific thought. In: *Selected writings on computing: a personal perspective*. Springer, Berlin, pp 60–66. ISBN 0-387-90652-5 29. OMG (2014) Meta object facility meta object facility (MOF) core specification. Adopted Specification, OMG document formal/14-04-03. <https://www.omg.org/spec/MOF/2.4.2> 30. Faieq S, Saidi R, Elghazi H, Rahmani MD (2017) A conceptual architecture for a cloud-based context-aware service composition. In: *Advances in ubiquitous networking 2*. Springer, Singapore, pp 235–246 31. Gu Q, Lago P (2009) Exploring service-oriented system engineering challenges: a systematic literature review. *Serv Orient Comput Appl* 3(3):171–188 32. Chengyuan Y, Huang L (2016) A web service qos prediction approach based on time- and location-aware collaborative filtering. *Serv Oriented Comput Appl* 10(2):135–149 33. Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: autoencoders meet collaborative filtering. In: *Proceedings of the 24th international conference on world wide web, WWW '15 Companion*. ACM, New York, NY, USA, pp 111–112 34. Stone VM (2008) The auto-associative neural network—a network architecture worth considering. In: *2008 world automation congress*, pp 1–4 35. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD*

international conference on management of data, SIGMOD'00. ACM, New York, NY, USA, pp 1–12 36. Garg K, Kumar D (2013) Comparing the performance of frequent pattern mining algorithms. *Int J Comput Appl* 69(25):21–28 37. WSDREAM team (2011) Ws-dream: A package of open sourcecode and datasets to benchmark QoS prediction approaches of web services 38. Zheng Z, Ma H, Lyu MR, King I (2011) Qos-aware web service recommendation by collaborative filtering. *IEEE Trans Serv Comput* 4(2):140–152 39. Wu J, Chen L, FengY, Zheng Z, Zhou MC,Wu Z (2013) Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Trans Syst Man Cybern Syst* 43(2):428–439 40. Sun H, Zheng Z, Chen J, LyuMR (2013) Personalized web service recommendation via normal recovery collaborative filtering. *IEEE Trans Serv Comput* 6(4):573–579 123 Service Oriented Computing and Applications 41. Zheng Z, Ma H, Lyu MR, King I (2013) Collaborative web service qos prediction via neighborhood integrated matrix factorization. *IEEE Trans Serv Comput* 6(3):289–299 42. Zhang Y, Zheng Z, Lyu MR (2011) Exploring latent features for memory-based qos prediction in cloud computing. In: 2011 IEEE 30th international symposium on reliable distributed systems, pp 1–10 43. Yu D, Liu Y, Xu Y, Yin Y (2014) Personalized QoS prediction for web services using latent factor models. In: 2014 IEEE international conference on services computing, pp 107–114 44. Chen X, Liu X, Huang Z, Sun H (2010) Regionknn: a scalable hybrid collaborative filtering algorithm for personalized web service recommendation. In: 2010 IEEE international conference on web services, pp 9–16 45. Tang M, Jiang Y, Liu J, Liu X (2012) Location-aware collaborative filtering for QoS-based service recommendation. In: 2012 IEEE 19th international conference on web services, pp 202–209 46. LoW, Yin J, Deng S, Li Y,Wu Z (2012) Collaborative web service GOS prediction with location-based regularization. In: 2012 IEEE 19th international conference on web services, pp 464–471 47. Chen X, Zheng Z, Yu Q, Lyu MR (2014) Web service recommendation via exploiting location and qos information. *IEEE Trans Parallel Distrib Syst* 25(7):1913–1924



48. He P, Zhu J, Zheng Z, Xu J, Lyu MR (2014) Location-based hierarchical matrix factorization for web service recommendation. In: 2014 IEEE international conference on web services, pp 297–304

49. Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization. In: Proceedings of the 20th international conference on neural information processing systems, NIPS'07. Curran Associates Inc, USA, pp 1257–1264

50. Lee Daniel D, Seung, H. Sebastian (2000) Algorithms for nonnegative matrix factorization. In: Proceedings of the 13th international conference on neural information processing systems, NIPS'00. MIT Press, Cambridge, MA, USA, pp 535–541

51. Rao J, SuX(2005)Asurvey of automated web service composition methods. In: Semantic web services and web process composition. Springer, Berlin, pp 43–54

52. Wu D, Parsia B, Sirin E, Hendler J, Nau D (2003) Automating DAML-S web services composition using shop 2. In: Proceedings of the second international conference on semantic web conference, LNCS-ISWC'03. Springer, Berlin, pp 195–210

53. Slaimi F, Hassine AB, Tagina M (2014) Ontology based vertical web service composition. *Int J Know Based Intell Eng Syst* 18(1):1–12

54. Furno A, Zimeo E (2014) Context-aware composition of semantic web services. *Mobile Netw Appl* 19(2):235–248

55. Chowdhury SR, Daniel F, Casati F (2011) Efficient, interactive recommendation of mashup composition knowledge. In: Serviceoriented computing. Springer, Berlin, pp 374–388

56. Budiseli'c I (2014) Component recommendation for development of composite consumer applications. PhD thesis, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu

57. Greenshpan O, Milo T, Polyzotis N (2009) Autocompletion for mashups. *Proc VLDB Endow* 2(1):538–549

58. Zhang S, Yao L, Sun A, Tay Y (2017) Deep learning based recommender system: a survey and new perspectives. *arXiv preprint [arXiv:1707.07435](https://arxiv.org/abs/1707.07435)*

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Adobe Acrobat  
Document