

# Построение адаптивного интерфейса цифровой библиотеки на основе онтологии данных

О.М. Атаева<sup>1</sup>, В.А. Серебряков<sup>1</sup>, Н.П.Тучкова<sup>1</sup>

<sup>1</sup>ФИЦ ИУ РАН, Москва, ул. Вавилова, 40

**Аннотация.** Работа посвящена проблеме настройки пользовательских интерфейсов информационной системы, осуществляющей интеграцию данных. Настраиваемый интерфейс служит одним из средств организации представления данных предметной области. Изучается вопрос об использовании семантических связей онтологии для подбора данных, соответствующих задачам исследований. Рассматривается модель адаптивного интерфейса, который позволяет наиболее точно отразить потребности исследователя в рамках определенной предметной области. Показано, как средствами, заложенными в модели семантической библиотеки, формируется адаптивный интерфейс.

**Ключевые слова:** онтология, адаптивный интерфейс, предметная область, модель данных

## Building an adaptive digital library interface based on data ontology

O.M.Ataeva<sup>1</sup>, V.A.Serebryakov<sup>1</sup>, N.P.Tuchkova<sup>1</sup>

<sup>1</sup>FRS «Computer Sciences and Control», Russian Academy of Sciences, Vavilov str., 40, Moscow, 119333, Russia

**Abstract.** The work is devoted to the problem of customizing the user interfaces of an information system that integrates data. An adaptive interface serves as one of the means of organizing the presentation of subject domain data. The issue of using the semantic relations of ontology to select data corresponding to the objectives of the study is investigated. A model of an adaptive interface is considered, which allows the most accurate reflection of the needs of a researcher within a particular subject domain. It is shown how the adaptive interface is formed by means of the semantic library model.

**Keywords:** ontology, adaptive interface, subject domain, data model

### Введение

Адаптивными информационными системами [1] (АИС) называют

системы, в которых заложены алгоритмы, меняющие поведение системы в ответ на действия пользователей или обладающие возможностью модифицировать свой контент не только на уровне данных, но и на уровне связей данных и, соответственно, модифицировать их представление пользователю. В рамках настоящей работы ограничимся рассмотрением интерфейса, настраиваемого в результате действия пользователя. Настройка интерфейса происходит на основе алгоритмов, заложенных в семантической библиотеке, как следствие работы пользователя в определенной предметной области (ПрО) интегрированной системы данных цифровой библиотеки. Адаптация системы на уровне настраиваемого интерфейса направлена на обеспечение информационной потребности пользователя при уменьшении информационного шума.

Идея адаптации интерфейсов информационных систем возникла естественным образом у многих разработчиков на этапе интеграции данных и необходимости обеспечить пользователя специфическим интерфейсом, характерным для определенных задач [1-3]. В процессе накопления данных в семантической библиотеке устанавливается множество связей. Связи необходимы для обеспечения полной информационной поддержки пользователей на базе контента информационной системы. Тем не менее, на определенном этапе пользователь может столкнуться с «лишней» информацией, которая не соответствует его интересу, т.е. pertinентности ответа системы. В этом случае система должна адекватно реагировать на запросы, аккумулируя информацию о пользователе, что и происходит в современных поисковых ресурсах. Развитие этого подхода привело к переходу от «монолитных», единых сервисов информационных систем к «микросервисам»[4-6].

В отношении научных знаний, представляемых в цифровых семантических библиотеках, адаптация интерфейсов системы связана с предоставлением пользователю информации из определенной ПрО. Если знания связаны с другими научными областями, приложениями, авторами, публикациями, то естественно, что информационная система должна «очертить область интересов» и предоставить пользователю возможность выбора информации при поиске.

В предлагаемой работе рассматривается адаптация интерфейса семантической библиотеки LibMeta<sup>1</sup> и ее математический контент, который опирается на классические источники академического сообщества такие, как математическая энциклопедия, классификаторы MSC<sup>2</sup> и УДК<sup>3</sup>, авторские тезаурусы<sup>4</sup> и словари.

---

<sup>1</sup> <https://libmeta.ru/>

<sup>2</sup> MSC2020-Mathematics Subject Classification System. <https://zbmath.org/static/msc2020.pdf>

<sup>3</sup> <https://teacode.com/online/udc/>

<sup>4</sup> ГОСТ 7.24-2007 Тезаурус информационно-поисковый многоязычный. <https://ifap.ru/library/gost/7242007.pdf>

## **АИС семантических библиотек.**

Выделяют несколько видов АИС [1, 2, 3]

- Системы, которые *в зависимости от действий пользователя* отображают соответствующую его информационным потребностям информацию. Это может достигаться за счет сложной организации ключевых слов, графических объектов, процессов, в которые вовлечен пользователь в информационной системе и т. д. Такие системы адаптируются под действия пользователя, варьируя как способ представления информации, так и саму информацию (например, разные пользователи перемещаются по списку найденной информации в разном порядке и это в дальнейшем влияет на то, какая информация будет им прежде всего отображаться, или отслеживается последовательность поисковых запросов чтобы предложить контекстную рекламу). В таких системах *модель данных жестко структурирована*.
- Обучающиеся системы, в основе которых лежат *методы автоматической классификации контента*, благодаря которым выстраивается определенная сеть иерархических и горизонтальных связей в поступающей информации. Система адаптируется под контент, а способы представления информации, и сама представляемая информация не зависят от внешних условий (например, действий пользователя). Отличительной особенностью таких систем является *возможность менять модель данных*, что отражается на классификации и представлении данных в системе.

Семантические библиотеки относятся [7-13] ко второму типу систем. С одной стороны, в таких библиотеках возникает необходимость ограничить контент в рамках некоторой ПрО. Для этого используется набор терминов, описывающих эту ПрО. Чаще всего эти термины организованы в виде некоторого тезауруса. С другой стороны, наполнение библиотеки представляет собой публикации, книги, проекты, задачи, т.е. разные ресурсы, перечень которых может меняться. Меняется также структура и связи этих ресурсов. Для тематической классификации ресурсов библиотеки могут использоваться различные классификаторы, которые отличаются друг от друга охватом ПрО и степенью детализации при классификации этих областей. То есть, можно сказать, что классификация ресурсов библиотеки основана на классификаторах и тезаурусе ПрО. При этом тезаурус может расширяться и пополняться

новыми понятиями, так же, как и классификатор.

Это выводит на передний план ряд проблем, связанных с реализацией семантической библиотеки, в частности:

- как изменить модель данных и отразить эти изменения в системе;
- как влияют эти изменения на представление данных в интерфейсах пользователей и других потребителей информации (имеются ввиду программные агенты, которые могут автоматически извлекать данные контента в машиночитаемом формате).

В целом, эти проблемы сводятся к двум вариантам моделей данных, и соответственно, подходам ее реализации.

Подход к разработке информационной системы семантической библиотеки *на предварительно жестко заданной модели данных* решает эти вопросы на этапе программной реализации, но любое изменение модели в таком случае требует дополнительной работы программистов. У подхода с жестко заданной моделью данных есть очевидные преимущества. В таких системах легче реализовать сложные взаимосвязи между ресурсами с одной стороны и проще построить «красивый» интерфейс для пользователя.

Подход, при котором *семантическая библиотека позволяет настраивать модель своего контента*, проще с точки зрения конечного пользователя, так как он получает возможность работы с моделью данных, не погружаясь в технические детали реализации. Это также ограничивает возможности моделирования «упрощая то, что можно упростить» в угоду возможности быстрой динамической настройки пользовательских интерфейсов под эти изменения.

В контексте этих проблем и на примере семантической библиотеки LibMeta в статье рассмотрим (1) что такое модель данных семантической библиотеки, (2) как настраивается модель данных, (3) как связаны изменения в модели данных с интерфейсами пользователей.

## **О модели данных библиотеки**

Для построения семантической библиотеки LibMeta используются семантические технологии из стека Semantic Web<sup>5</sup>. Особое значение имеют онтологии<sup>6</sup>, которые позволяют составить модель данных на основе ее понятий и отношений между ними. Понятия онтологии семантической

---

<sup>5</sup> <https://www.w3.org/standards/semanticweb>

<sup>6</sup> <https://www.w3.org/standards/semanticweb/ontology>

библиотеки можно разделить условно на 2 группы:

- понятия (первого уровня), которые дают высокоуровневый взгляд на контент библиотеки (например: ресурс, атрибут, тезаурус);

- понятия (второго уровня) конкретной ПрО (задачи математической физики, автор публикаций, тезаурус Обыкновенных Дифференциальных Уравнений (ОДУ), ... ).

Сами данные (например, задача Коши, Иванов И.И., ...) представляются на основе заданных понятий.

*Понятия первого уровня* предоставляют возможность проектировать и реализовывать программный интерфейс семантической библиотеки, который, в свою очередь, позволяет описывать *понятия второго уровня*. Они составляют *Тезаурус первого уровня*. Например, для того чтобы семантическая библиотека соответствовала ПрО «Математика», вводятся такие понятия как «Публикация», «Персона», «Формула», которые связываются с тезаурусом ОДУ и «Математической энциклопедией». Математическая энциклопедия также является *экземпляром понятия «Тезаурус первого уровня»*. То есть, в основе пользовательских интерфейсов настройки ПрО лежит *модель данных понятий первого уровня*.

На Рис. 1 представлен в общем виде процесс формирования интерфейсов библиотеки на основе модели данных. Для этого процесса необходимо выполнить следующую последовательность шагов:

- Определить множество ресурсов (*Публикация, Автор, ...*);
- Определить атрибуты (*название публикации, код (Номер) MSC, является автором, ...*);
- Сформировать множество атрибутов для каждого ресурса;
- Сформировать интерфейсы системы на основе видов атрибутов:
  - идентификационные атрибуты — используются при выводе краткой информации в различных формах и при поиске дубликатов;
  - поисковые атрибуты — используются при формировании форм атрибутного поиска;
  - описательные атрибуты — используются для формирования форм редактирования и просмотра.

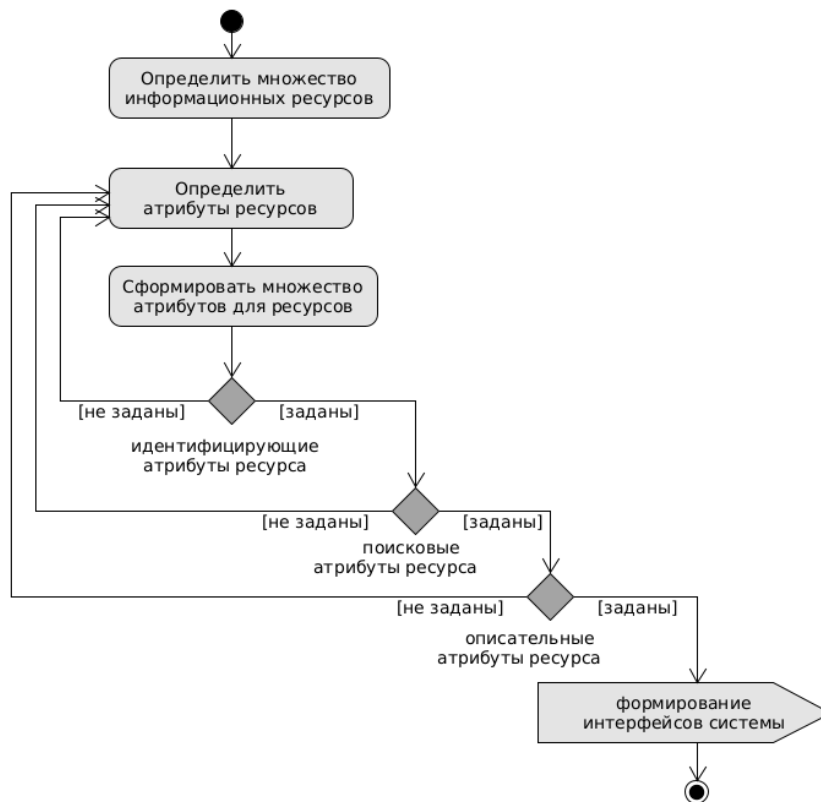


Рис. 1. Настройка интерфейсов

## Пользовательские интерфейсы

Для удобства пользователей необходимо предоставить возможность поиска, просмотра и редактирования информации средствами библиотеки. Для этого при описании ПрО определяются типы и виды атрибутов. Типы это: *текст*, *число*, *объект*, *понятие*, *ссылка*. Задание типа атрибута позволяет выделить для него соответствующий элемент в формах пользователя и обеспечивает навигацию между связанными ресурсами.

Рассмотрим пример настройки интерфейсов (Рис. 2-6) при определении атрибута «Номер MSC», который связывает публикации с классификатором MSC, при этом соответствующий ресурс *Публикация* уже предварительно был создан. Понятия *Ресурс* и *Атрибут* являются понятиями первого уровня. На Рис. 2 показан фрагмент онтологии, в котором отображены понятия, связанные с понятием *Атрибут*. На основе этого фрагмента онтологии в форме описания атрибута (Рис. 3) при указании атрибута «Номер MSC» и его типа «Таксономия», используя конкретный классификатор MSC на форме редактирования публикации (Рис. 4), получим элемент, который позволяет найти и «привязать» элемент классификатора к конкретной публикации, а на форме просмотра (Рис. 5)

появится возможность перейти по ссылке и посмотреть связанные с ним объекты.

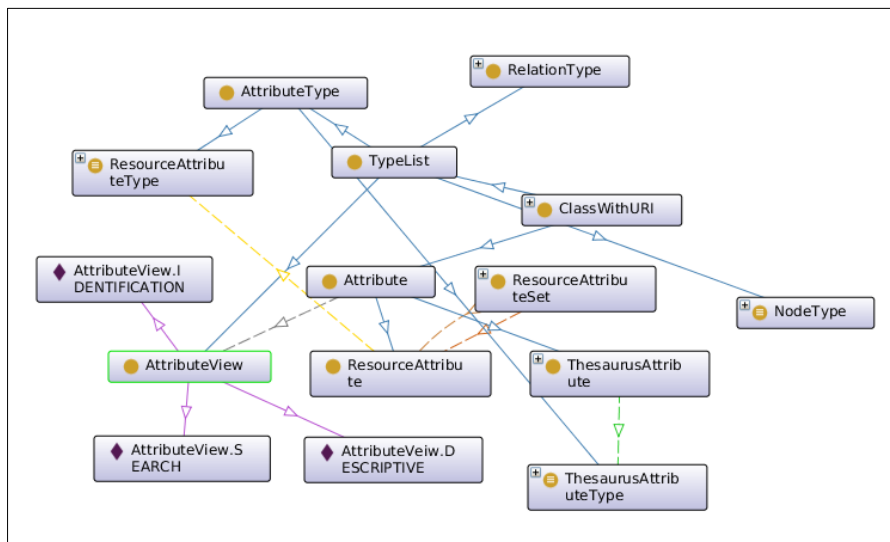


Рис. 2. Фрагмент онтологии

**Редактировать атрибут**

Название (на русск.) \*

Вид представления  - Поисковый  - Идентифицирующий  - Описательный

Многозначный \*

Видимость \*

Тип значений \*

Тип значений объектов

Рис. 3. Форма описания атрибута

**Редактировать объект**

Тип объекта \*

Атрибуты

Название

Номер MSC [Выбрать элементы](#)

[00A08 - Удалить](#)

Рис. 4. Форма редактирования публикации

Просмотр объекта	
<b>Бесконечно мелкие разбиения пространств с мерой</b>	
Тип объекта	<a href="#">Публикация</a>
Атрибуты	<a href="#">Название</a> - Бесконечно мелкие разбиения пространств с мерой <a href="#">Номер MSC</a> - <a href="#">00A08 - Recreational mathematics</a> <a href="#">связанные объекты</a> <a href="#">связанные концепты</a>

Рис. 5. Форма просмотра публикации

Поиск информационных объектов	
Тип объекта *	<input type="text" value="Публикация"/>
Атрибуты	
Название на русском	<input type="text"/>
Номер MSC	<a href="#">Выбрать элементы</a>
Журнал	<input type="text"/>

Рис. 6. Поисковая форма

Виды атрибута *поисковый*, *описательный*, *идентифицирующий* в свою очередь позволяют указать, в каком виде представления информации о ресурсе тот или иной атрибут участвует. Так как атрибут «Номер MSC» отмечен как *поисковый*, то соответствующий элемент появляется на *поисковой форме* (Рис. 6).

## Программный интерфейс REST API

Все настройки модели данных средствами библиотеки отражаются не только на интерфейсах пользователя, но также и на общедоступном программном интерфейсе<sup>7</sup> (Application Programming Interface, API), который позволяет удаленно обращаться к данным библиотеки. Такой прикладной интерфейс был реализован на технологии REST<sup>8</sup> (Representational State Transfer – передача представления состояния).

Рассмотрим в качестве примера объект класса *Публикация*, который является отдельным элементом контента библиотеки и который можно извлечь, изменить или удалить. На Рис. 7 представлены ключевые понятия

<sup>7</sup> <https://www.openapis.org/>

<sup>8</sup> [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)



технологии REST: информационный объект, который является экземпляром информационного ресурса ПрО, и представление состояния информационного объекта, т.е. описание состояния объекта в терминах онтологии ПрО, возвращаемый клиенту в RDF/XML разметке.

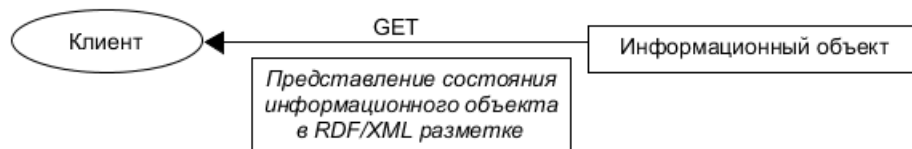


Рис. 7. Ключевые понятия REST API

Точкой входа для доступа к RESTfull API в нашей системе является URL вида: `http(s)://[имя сервера]/rest/`. Просмотреть описание основных элементов модели данных, на основе которых формируется настраиваемый интерфейс, можно используя методы группы `/resource`. На Рис. 8 представлен фрагмент такого описания в формате RDF/XML в LibMeta.

```

    https://libmeta.ru/resources x +
    libmeta.ru/resources
    <?xml version="1.0" encoding="UTF-8" ?>
    <!DOCTYPE InformationResource [
    <rdf:RDF xmlns:lbm="http://libmeta.ru/resource/publication"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    >
    <lbm:InformationResource rdf:about="http://libmeta.ru/resource/publication">
    <lbm:title>Publication</lbm:title>
    <lbm:label>Публикация</lbm:label>
    <lbm:description>Ресурс соответствующий публикациям</lbm:description>
    <lbm:properties>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#aboutTitle"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#annt"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#auth"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#bibliography"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#desc"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#full"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#issueDate"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#issueMonth"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#keywords"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#lang"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#msc"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#magazine"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#media"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#originalTitle"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#udc"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#pubtype"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#riopubtype"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#russianTitle"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#seeAlso"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#source"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#storage"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#tomNumber"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute/category"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute/formula"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute/href"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute/pages"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute/pubhouse"/>
    <lbm:property rdf:resource="http://libmeta.ru/attribute#vipNumber"/>
    </lbm:properties>
    <lbm:dateCreated> 01-11-2018 02:34 </lbm:dateCreated>
    <lbm:dateUpdated> 01-11-2018 02:34 </lbm:dateUpdated>
    </lbm:InformationResource>
    </rdf:RDF>
    ]
  
```

Рис. 8. Фрагмент модели данных в формате RDF/XML

## Заключение

В модели семантической библиотеки введены понятия для описания содержимого библиотеки некоторой ПрО. Эти понятия позволяют сконструировать описание любых типов информационных ресурсов для выбранной области в рамках контента библиотеки. Информационные объекты, являющиеся непосредственно содержимым библиотеки, имеют распределенную природу, а именно, данные могут поступать из различных источников и агрегировать информацию из различных источников, что приводит к изменениям в существующей модели и их соответствующее отображение в интерфейсах библиотеки. Для реализации этого алгоритма как нельзя лучше подходит построение *адаптивных* интерфейсов системы на основе *модели данных* описания научных ресурсов. Это позволяет не ограничиваться при разработке строго очерченным набором ресурсов. Применение адаптивной модели позволяет понизить сложность (размерность) как самой модели данных, так и разрабатываемых на ее основе систем, ускоряя внедрение и разработку семантической библиотеки.

Работа представлена в рамках выполнения темы госзадания «Математические методы анализа данных и прогнозирования» ФИЦ ИУ РАН и частично поддержана проектом РФФИ # 20-07-00324

## Литература

1. Sabatucci L., Seidita V., Cossentino M. The Four Types of Self-adaptive Systems: A Metamodel. In: De Pietro, G., Gallo, L., Howlett, R., Jain, L. (eds) Intelligent Interactive Multimedia Systems and Services 2017. KES-ИИМСС-18 2018. Smart Innovation, Systems and Technologies, 2018, vol. 76, pp. 440–450. Springer, Cham. [https://doi.org/10.1007/978-3-319-59480-4\\_44](https://doi.org/10.1007/978-3-319-59480-4_44).  
[https://www.researchgate.net/publication/318132984\\_The\\_Four\\_Types\\_of\\_Self-adaptive\\_Systems\\_A\\_Metamodel](https://www.researchgate.net/publication/318132984_The_Four_Types_of_Self-adaptive_Systems_A_Metamodel).
2. Ferreira H., Correia F. and Aguiar A. Design for an Adaptive Object-Model Framework. An Overview. Proceedings of the 4th Workshop on Models@run.time, held at the ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MoDELS'09) Denver, USA, October 5th, 2009. // *CEUR Workshop Proceedings*, M. Jeusfeld c/o Redaktion Sun SITE, Informatik V, RWTH Aachen (Aachen, Germany), 2009, vol. 509, pp. 71-80. [http://ceur-ws.org/Vol-509/paper\\_13.pdf](http://ceur-ws.org/Vol-509/paper_13.pdf).

3. Yoder J. W., Johnson R. The Adaptive Object-Model Architectural Style. 2002. <https://www.researchgate.net/publication/220864957>.
4. Fowler M. and Lewis J. “Microservices”. 2014. [Online]. <http://martinfowler.com/articles/microservices.html>.
5. Andrade B., Santos S. and Silva A.R. From Monolith to Microservices: Static and Dynamic Analysis Comparison. 2022. <https://10.48550/arXiv.2204.11844> Corpus ID: 248392322.
6. Santos N. and Silva A.R. A Complexity Metric for Microservices Architecture // Computer Science. IEEE International Conference on Software Architecture (ICSA) 2020.
7. Mascardi V., Cordi V., Rosso P. A Comparison of Upper Ontologies. Conference: WOA 2007: Dagli Oggetti agli Agenti. 8th AI\*IA/TABOO Joint Workshop "From Objects to Agents": Agents and Industry: Technological Applications of Software Agents, 24-25 September 2007, Genova, Italy. 2007.
8. Katsis Y., Papakonstantinou Y. View-based data integration //Encyclopedia of Database Systems. 2009. C. 3332-3339.
9. Xu L., Embley D. W. Combining the Best of Global-as-View and Local-as-View for Data Integration //ISTA. 2004. T. 48. C. 123-36.
10. Noy N. F. Semantic integration: a survey of ontology-based approaches //ACM Sigmod Record. 2004. T. 33. №. 4. C. 65-70.
11. Zhao L., Ichise R. Ontology integration for linked data // Journal on Data Semantics. 2014. V. 3. №. 4. S. 237-254.
12. Serebryakov V. A., Ataeva O. M. Ontology Based Approach to Modeling of the Subject Domain “Mathematics” in the Digital Library //Lobachevskii Journal of Mathematics. 2021. V. 42. №. 8. S. 1920-1934. <https://mathscinet.ams.org/msc/msc2010.html>.
13. Ataeva O. M., Serebryakov V. A., Tuchkova N. P. Creation of query expansion based on the subject domain thesaurus in the ontology of knowledge of the semantic library // Scientific service on the Internet: Proceedings of the XXI All-Russian Scientific Conference (September 23-28, 2019, Novorossiysk). <https://doi.org/10.20948/abrau-2019-12>.

## References

1. Sabatucci L., Seidita V., Cossentino M. The Four Types of Self-adaptive Systems: A Metamodel. In: De Pietro, G., Gallo, L., Howlett, R., Jain, L. (eds) Intelligent Interactive Multimedia Systems and Services 2017. KES-IIMSS-18 2018. Smart Innovation, Systems and Technologies, 2018, vol. 76, pp. 440–450 Springer, Cham. [https://doi.org/10.1007/978-3-319-59480-4\\_44](https://doi.org/10.1007/978-3-319-59480-4_44).  
[https://www.researchgate.net/publication/318132984\\_The\\_Four\\_Types\\_of\\_Self-adaptive\\_Systems\\_A\\_Metamodel](https://www.researchgate.net/publication/318132984_The_Four_Types_of_Self-adaptive_Systems_A_Metamodel).

2. Ferreira H., Correia F. and Aguiar A. Design for an Adaptive Object-Model Framework. An Overview. Proceedings of the 4th Workshop on Models@run.time, held at the ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MoDELS'09) Denver, USA, October 5th, 2009. // *CEUR Workshop Proceedings*, M. Jeusfeld c/o Redaktion Sun SITE, Informatik V, RWTH Aachen (Aachen, Germany), 2009. V. 509. P. 71-80. [http://ceur-ws.org/Vol-509/paper\\_13.pdf](http://ceur-ws.org/Vol-509/paper_13.pdf).
3. Yoder J. W., Johnson R. The Adaptive Object-Model Architectural Style. 2002. <https://www.researchgate.net/publication/220864957>.
4. Fowler M. and Lewis J., "Microservices". 2014. [Online]. <http://martinfowler.com/articles/microservices.html>.
5. Andrade B., Santos S. and Silva A.R. From Monolith to Microservices: Static and Dynamic Analysis Comparison. 2022. <https://10.48550/arXiv.2204.11844> Corpus ID: 248392322.
6. Santos N. and Silva A.R. A Complexity Metric for Microservices Architecture // Computer Science. IEEE International Conference on Software Architecture (ICSA) 2020.
7. Mascardi V., Cordi V., Rosso P. A Comparison of Upper Ontologies. Conference: WOA 2007: Dagli Oggetti agli Agenti. 8th AI\*IA/TABOO Joint Workshop "From Objects to Agents": Agents and Industry: Technological Applications of Software Agents, 24-25 September 2007, Genova, Italy. 2007.
8. Katsis Y., Papakonstantinou Y. View-based data integration // Encyclopedia of Database Systems. 2009. S. 3332-3339.
9. Xu L., Embley D. W. Combining the Best of Global-as-View and Local-as-View for Data Integration // ISTA. 2004. V. 48. S. 123-36.
10. Noy N. F. Semantic integration: a survey of ontology-based approaches. // ACM Sigmod Record. 2004. V. 33. №. 4. S. 65-70.
11. Zhao L., Ichise R. Ontology integration for linked data // Journal on Data Semantics. 2014. V. 3. №. 4. S. 237-254.
12. Serebryakov V. A., Ataeva O. M. Ontology Based Approach to Modeling of the Subject Domain "Mathematics" in the Digital Library // Lobachevskii Journal of Mathematics. 2021. V. 42. №. 8. S. 1920-1934. <https://mathscinet.ams.org/msc/msc2010.html>.
13. Ataeva O. M., Serebryakov V. A., Tuchkova N. P. Creation of query expansion based on the subject domain thesaurus in the ontology of knowledge of the semantic library // Scientific service on the Internet: Proceedings of the XXI All-Russian Scientific Conference (September 23-28, 2019, Novorossiysk). <https://doi.org/10.20948/abrau-2019-12>.