

УДК 004.896

## ПРИМЕНЕНИЕ МЕТОДА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ РАЗРАБОТКЕ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ МУЛЬТИАГЕНТНОГО ПОДХОДА

**Аксенов К.А., Спицина И.А.**

*ФГАОУ ВПО «Уральский федеральный университет им. первого Президента России  
Б.Н. Ельцина», Екатеринбург, e-mail: wiper99@mail.ru*

В работе описан пример использования метода поддержки принятия решений при разработке информационных систем микросервисной архитектуры для проектирования веб-сервиса. Метод опирается на гибридный подход в проектировании, который заключается в совместном использовании структурного анализа и объектно-ориентированного подхода: на этапе обследования используются нотации описания процессов IDEF0 и DFD; на этапе проектирования информационной системы диаграмма DFD конвертируется в элементы (артефакты) диаграмм классов, прецедентов и последовательности языка UML, что обеспечивает согласованность и корректность выполнения данных этапов. Он позволяет описывать автономные и автоматизированные компоненты программных систем, а также веб-сервисы как программных агентов. Взаимодействие с агентом идет с использованием методов API. Эти методы могут быть показаны на диаграмме последовательности языка UML, описывающей работу агента-микросервиса. Информация для построения диаграмм объектно-ориентированного подхода берется из описания диаграмм структурного подхода. Для апробации метода был разработан прототип веб-сервиса имитационного моделирования BPsim.Web. Основные функции веб-сервиса имитационного моделирования BPsim.Web – построение гибридных агентных имитационных дискретно-событийных моделей и проведение имитационных экспериментов с моделями бизнес-процессов и организационно-технических систем, анализ результатов имитационных экспериментов.

**Ключевые слова:** информационная система, мультиагентный подход, микросервисная архитектура, имитационное моделирование, веб-сервис

## APPLICATION OF THE DECISION SUPPORT METHOD FOR THE DEVELOPMENT OF INFORMATION SYSTEMS BASED ON THE MULTI-AGENT APPROACH

**Aksenov K.A., Spitsina I.A.**

*Ural Federal University named after First President of Russia B.N. Yeltsin, Ekaterinburg,  
e-mail: wiper99@mail.ru*

The article describes an example of using the decision support method in the development of multiservice architecture information systems for a web service. The method is based on a hybrid approach, which consists in the joint use of structural analysis and an object-oriented approach. For domain description stage the method are used IDEF0 and DFD sequence diagram notation. On the software design stage DFD diagram are converted in artefact and elements of classes diagram, use cases diagram and sequences diagram. This approach are procuring the correction of both stages. It allows you to describe autonomous and automated components of software systems, as well as web services as software agents. Interaction with the agent is carried out using API methods. These methods can be shown on a UML sequence diagram describing the operation of a microservice agent. The information for constructing diagrams of the object-oriented approach is taken from the description of diagrams of the structural approach. To test the method, a prototype of the web service BPsim was developed. The main functions of the BPsim.Web service is the construction of hybrid agent-based simulation discrete-event models and conducting simulation experiments with models of business processes and organizational and technical systems, analysis of the results of simulation experiments.

**Keywords:** information system, multi-agent approach, microservices architecture, simulation modeling, web service

Разработка программного обеспечения (ПО) представляет собой процесс, сложность которого определяется уровнем понимания используемых современных технологий и автоматизируемых бизнес-процессов. На каждом этапе команде разработчиков необходимо принимать решения при наличии сложной, во всегда полной и разносторонней информации. Поэтому возникают проектные риски: управленческие, функциональные и технические. Первые связаны с принятием решений, касающихся проектной команды. Функциональные риски возникают из-за изменений в бизнес-требо-

ваниях. Третьи – с выбором используемых информационных технологий, архитектуры программного обеспечения. Успешность проекта во многом зависит от успешности управления рисками – выявление рисков и разработка мероприятий по их недопущению. Использование формализованных методов описания проектных решений и методов проектирования ПО (например, структурный или объектно-ориентированный) позволяет снизить риски и повысить качество разрабатываемых систем.

Основной целью исследования является совершенствование метода агентно-

ориентированного проектирования информационных систем для предметной области процессов преобразования ресурсов и его адаптация при проектировании веб-сервисов.

### Материалы и методы исследования

Важный вопрос на начальном этапе разработки информационной системы (ИС) – выбор архитектуры. Исторически первой появилась монолитная архитектура. ИС, разработанная по этой концепции, представляет собой единое приложение, которое состоит из пользовательского интерфейса, серверного приложения и базы данных. Все части ИС взаимосвязаны и взаимозависимы. Это имеет свои достоинства и недостатки. К положительным моментам относятся: простое и быстрое развертывание приложения, минимизация проблем, связанных со взаимодействием компонент. К недостаткам можно отнести: сложность внедрения новых информационных технологий, также в результате доработок ИС код приложения становится громоздким и трудным для понимания [1].

Развитие информационных технологий привело к появлению новой сервис-ориентированной архитектуры (SOA). ИС, разработанная в соответствии с этой архитектурой, представляет собой совокупность отдельных модулей, взаимодействующих друг с другом по стандартизованному протоколу. SOA разделяет компоненты по двум основным ролям: поставщик и потребитель сервисов. Обе эти роли могут играть программные агенты [1]. Плюсами этой архитектуры являются: повторное использование отдельных модулей в других проектах, независимость модулей приводит к более простому внесению доработок и развитию ИС. Основная проблема этой архитектуры – управление обменом сообщений между сервисами [1].

Микросервисная архитектура стала развитием сервисно-ориентированной. Приложение состоит из автономных компонентов, которые взаимодействуют между собой посредством API (Application Programming Interface) [2].

### Результаты исследования и их обсуждение

#### *Метод проектирования ИС на основе агентного подхода*

Организационно-техническая система (ОТС) представляет собой совокупность организационной структуры и находящихся в ее распоряжении технических средств, т.е. совместно рассматривается человек

и информационная система. Современные тенденции информатизации – не просто автоматизация бизнес-процессов предприятия, а создание виртуальной организации, охватывающей всех участников процесса. Поскольку для ОТС характерны процессы принятия решений (знания, сценарии, согласования), то метод проектирования таких систем должен включать в себя формализацию и информатизацию процессов принятия решений (ППР). Лица, принимающие решения, могут рассматриваться как интеллектуальные агенты. Таким образом, ОТС может быть рассмотрена как мультиагентная система (МАС). Существующий метод ППР при разработке информационных систем состоит из нескольких этапов [3].

Первый этап – обследование процессов ОТС. На данном этапе проводится обследование предметной области. Затем, на основании полученных данных строится имитационная модель «как есть» деятельности обследуемого предприятия. При этом используется модель мультиагентных процессов преобразования ресурсов, которая основана на интеграции имитационного, экспертного, ситуационного и мультиагентного моделирования [4, 5].

Второй этап – проводятся имитационные эксперименты с моделью «как есть» с целью выявления «узких мест» в организации процессов. По результатам моделирования строится динамическая модель мультиагентных процессов преобразования ресурсов (МППР) [4, 5] «как будет». К основным элементам модели МППР относятся: ресурсы, заявки, средства, операции, агенты.

Далее, переходим к следующему этапу – проработка проекта. Отличительной особенностью предлагаемого метода является возможность использовать данные модели процессов ОТС для построения модели ИС в нотациях IDEF0 и DFD. Дополнительно, на данном этапе при построении модели ИС предложено использовать результаты структурного анализа, как основы для объектно-ориентированного проектирования.

В ходе дальнейшего объектно-ориентированного проектирования с использованием языка UML достигается построение полной модели, проектируемой ИС, включающей в себя компоненты мультиагентной имитационной модели МППР. На основании модели ИС получаем заготовки программных модулей и переходим к третьему этапу – разработке системы. Для формализации и реализации микросервисной архитектуры можно использовать концепцию МППР, поскольку она включает в себя агентов, ресурсы и операции.

Понятие «агент» соответствует аппаратно или программно реализованной сущности, которая способна действовать в интересах достижения целей, поставленных перед ней владельцем и/или пользователем, обладающей интеллектуальными способностями [6–9].

Мир агента составляет его окружение: 1) ресурсы, средства, заявки, за которыми он наблюдает; 2) операции и агенты, которыми он управляет; 3) агенты с которыми он обменивается информацией и взаимодействует. С точки зрения проектируемой ИС с помощью агентов могут быть формализованы модели поведения пользователей или отдельных компонент программной системы.

На этапе формализации предметной области с помощью модели МППР могут быть формализованы как программные компоненты (не только автономные), но и модели поведения пользователей. Вопросы модели и агентно-ориентированного проектирования освещаются в работах В.А. Виттиха и П.О. Скобелева [10, 11], Б.В. Соколова [12]. Концептуальная модель предметной области МППР включает в себя Агентов [3]. При переходе к модели предметной об-

ласти ИС агент преобразуется в объекты диаграмм DFD и UML (рис. 1).

Если рассмотреть некоего программного агента в нотации DFD, то выполняемые функции агентом будут представлены в виде подмножества функций DFD, переменные представляющие наблюдаемые переменные или формируемые сообщения – представляющие формируемые сообщения – представляют подмножества хранилищ на диаграмме DFD, а сам агент с точки зрения взаимодействия с пользователями и другими агентами – внешнюю сущность.

На рис. 1 показано, что агент представлен на DFD-диаграмме в виде Хранилища и Функции или Функций, которые он выполняет. На основе результатов структурного анализа строится UML-диаграмма прецедентов, на которой агенту сопоставляется определенная роль, и показаны связанные с ним прецеденты. Также эти данные могут быть использованы для формирования класса агента.

При проектировании ИС в архитектуре SOA или микросервисной отдельные сервисы могут быть рассмотрены в виде программных агентов, работающих автономно и/или взаимодействующих с конечными пользователями.

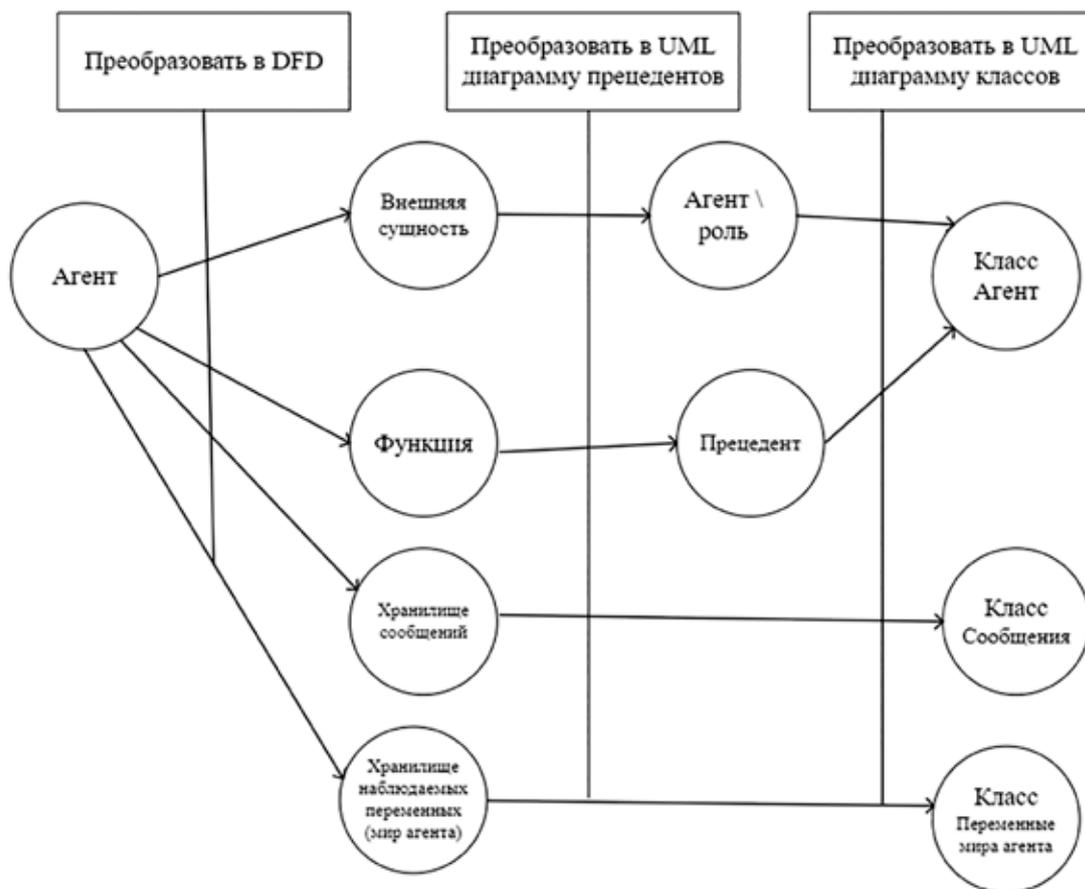


Рис. 1. Семантика перехода агента в объекты диаграмм

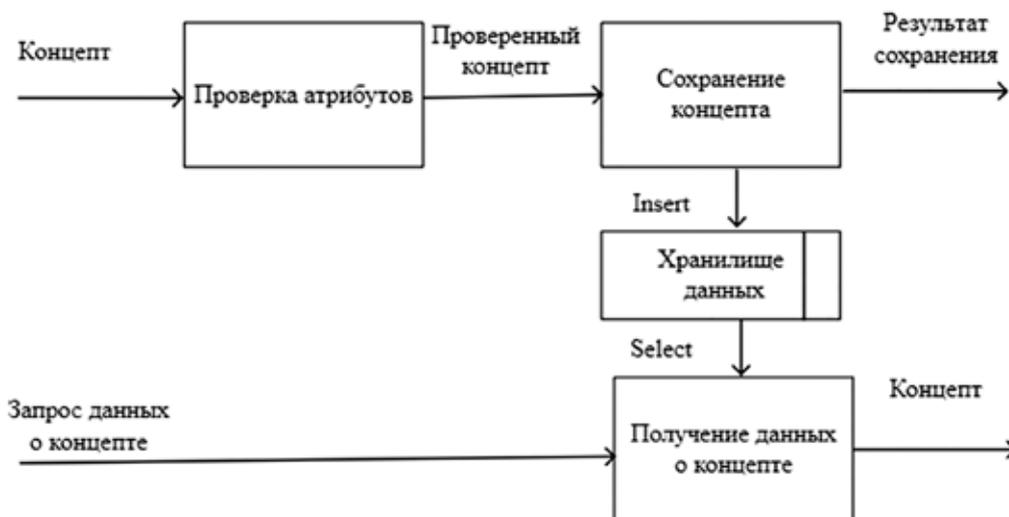


Рис. 2. DFD-диаграмма работы агента-микросервиса

Рассмотрим применение нашего метода при проектировании микросервиса, который реализует определенную функцию. При проектировании методов RESTful-API необходимо определить, какие данные необходимо предоставлять другим приложениям. В RESTful-API существует четыре возможности работы с информационными концептами: создать, удалить, получить и изменить. Пусть наш микросервис создаст некоторую сущность и позволяет получить данные о ней, тогда у агента есть два метода: POST и GET.

Фрейм-концепт «Агент» будет иметь следующие свойства: идентификатор, на-

звание агента, цели агента, приоритет агента, количество входящих сообщений, количество исходящих сообщений и методы: сохранить ресурс (POST), получить информацию о ресурсе (GET). Если агент оперирует неким подмножеством информационных концептов, то соответствующие методы требуются для каждого из них.

Рассмотрим преобразование агента-микросервиса в объекты концептуальной модели предметной области (КМПО) ИС. Для хранения информации о ресурсах на DFD-диаграмме создается хранилище данных. На рис. 2 показана работа агента-микросервиса с ресурсом.

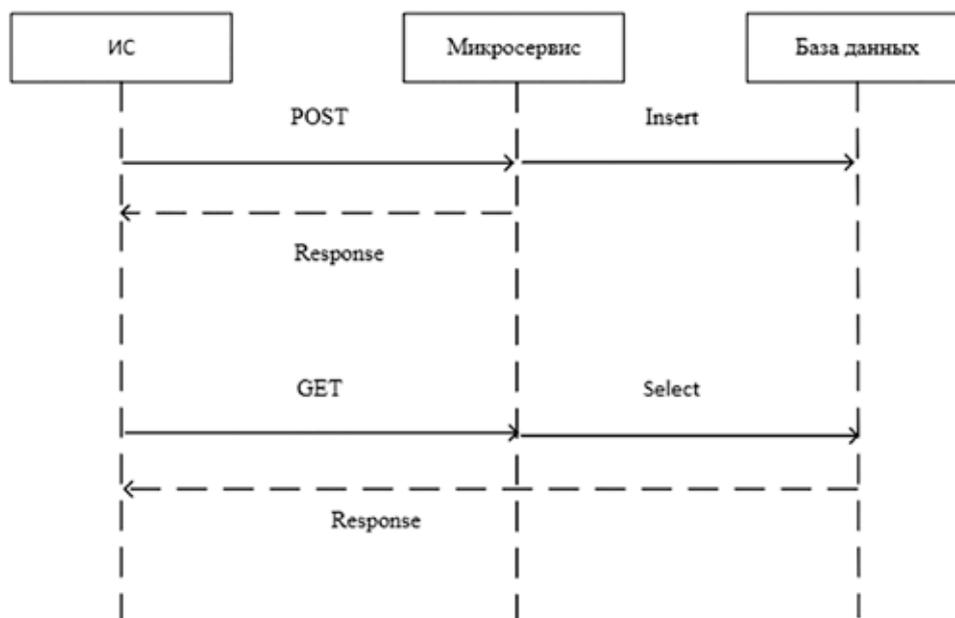


Рис. 3. UML-диаграмма последовательности работы агента-микросервиса

Таблица 1

## Описание методов веб-сервиса BPsim.Web

Метод	Название	Пояснение
POST	/model	Создать модель
GET	/model	Получить данные обо всех моделях
GET	/model/{id}	Получить данные о конкретной модели
DELETE	/model/{id}	Удалить данные о конкретной модели
PUT	/model/{id}	Изменить данные о конкретной модели
POST	/model/{id}/task	Создать задачу на проведение имитационного эксперимента
GET	/task	Получить данные обо всех задачах на выполнение имитации
GET	/task/{id}	Получить данные о конкретной задаче на выполнение имитации
DELETE	/task/{id}	Удалить данные о конкретной задаче на выполнение имитации
GET	/task/{id}/report	Получить отчеты о результатах выполнения конкретной задачи на выполнение имитации

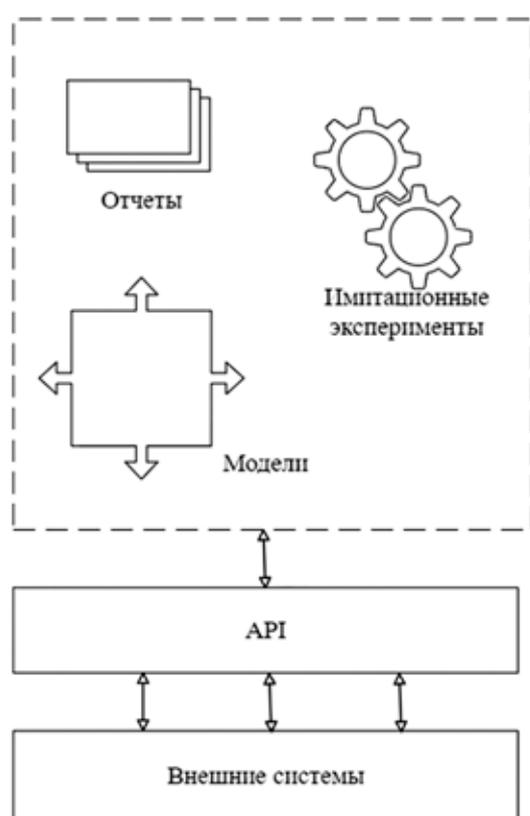


Рис. 4. Архитектура сервиса имитационного моделирования бизнес-процессов

Для описания методов RESTful-API можно использовать спецификацию OpenAPI. При проектировании диаграммы последовательности можно описать тип метода и его параметры.

Для проверки адекватности метода в контексте проектирования ИС на основе микросервисов был разработан прототип

сервиса имитационного моделирования BPsim.Web. Он представляет собой высоконагруженный веб-сервис, который позволяет работать с мультиагентными имитационными моделями бизнес-процессов.

На рис. 4 представлена архитектура сервиса имитационного моделирования.

Сервис управляет двумя концептами: модель имитационного моделирования и задача на выполнение имитации. Он получает команды из интерфейса интеграции (API) и, в зависимости от команды, получает или сохраняет данные в базу данных, выполняет внутренние преобразования и расчеты, запускает имитационный эксперимент. В ходе проектирования сервиса имитационного моделирования BPsim.Web были разработаны методы, представленные в табл. 1.

Объект модель БП имеет следующую структуру, представленную в табл. 2.

Таблица 2

## Структура модели бизнес-процесса

Название атрибута	Пояснение
name	Название модели
resources	Список ресурсов
orders	Список заявок
nodes	Список узлов (агенты, операции)

Объекты ресурс, заявка и узел имеют следующую структуру: список свойств и их значений.

Разработанный прототип веб-сервиса позволяет осуществлять следующие действия через API:

– Создавать, получать, изменять и удалять имитационные модели.

– Планировать и запускать имитационные прогоны.

– Выгружать информацию о результатах имитационного моделирования.

Используя разработанный сервис, были построены две имитационные модели: модель отделов сопровождения телекоммуникационных систем цехов прокатки и модель взаимодействия цехов холодной и горячей прокатки. По каждой из моделей был проведен набор экспериментов и решена задача оптимизации.

### Заключение

Предложенный авторами метод поддержки принятия решений при разработке информационных систем на основе мультиагентного подхода может быть использован при разработке веб-сервисов. Агент МППР подходит для описания веб-сервиса, доработка пакета VPsim позволяет описывать API-методы сервиса. Разработанный прототип сервиса имитационного моделирования VPsim.Web в ходе проведения имитационных экспериментов показал свою работоспособность.

### Список литературы

1. Лучшая архитектура для MVP: монолит, SOA, микросервисы или бессерверная?.. Ч. 1 [Электронный ресурс]. URL: <https://habr.com/ru/company/otus/blog/476024/> (дата обращения: 27.10.2022).
2. Лучшая архитектура для MVP: монолит, SOA, микросервисы или бессерверная?.. Ч. 2 [Электронный ресурс]. URL: <https://habr.com/ru/company/otus/blog/477930/> (дата обращения: 27.10.2022).
3. Спицина И.А., Аксенов К.А. Метод поддержки принятия решений при разработке информационных систем на основе мультиагентного подхода: монография. Министерство науки и высшего образования Российской Федерации, ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»; ФГБОУ ВО «Уральский государственный педагогический университет». Екатеринбург: УрГПУ, 2018. 156 с.
4. Antonova A.S., Aksyonov K.A., Aksyonova O.P. An imitation and heuristic method for scheduling with subcontracted resources. *Mathematics*. 2021. No. 9 (17). 2098. DOI: 10.3390/math9172098.
5. Antonova A., Aksyonov K., Ziomkovskaya P. (2021) Development of a Method and a Software for Decision-Making, System Modeling and Planning of Business Processes. *Frontiers in Software Engineering. ICFSE 2021. Communications in Computer and Information Science*. Vol. 1523. Springer. Cham. 2021. DOI: 10.1007/978-3-030-93135-3\_10.
6. Wooldridge M., Jennings N., Kinny D. The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems* 3. 2000. P. 285–312.
7. Wooldridge M., Jennings N. *Intelligent Agent: Theory and Practice*. Knowledge Engineering Review. 1995. № 10 (2).
8. Швецов А.Н. Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений: дис. ... докт. техн. наук: 05.13.01. Санкт-Петербург, 2004. 461 с.
9. Андрейчиков А.В., Андрейчикова О.Н. *Интеллектуальные информационные системы: учебник*. М.: Финансы и статистика, 2004. 424 с.
10. Skobelev P., Zhilyaev A., Larukhin V., Grachev S., Simonova E. Ontology-based open multi-agent systems for adaptive resource management. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*. Valletta, Malta. 2020. P. 127–135. DOI: 10.5220/0008896301270135.
11. Skobelev P., Zhilyaev A., Larukhin V., Grachev S., Simonova E. Ontology-based open multi-agent systems for adaptive resource management. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*. Valletta, Malta. 22–24 February 2020. P. 127–135. DOI: 10.5220/0008896301270135.
12. Lukinskiy V., Lukinsky V., Sokolov B., Bazhina D.A. Probabilistic Estimation of Perfect Order Parameters, in: *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021*. Nantes, France, September 5–9, 2021. *Proceedings Part 1*. Switzerland: Springer, 2021. P. 447–454.