

Научная статья
УДК 004.9
doi:10.37614/2949-1215.2025.16.3.002

ПОДХОД К ФОРМИРОВАНИЮ ОНТОЛОГИИ ЦИКЛА УПРАВЛЕНИЯ ЖИЗНЕСПОСОБНОСТЬЮ КРИТИЧЕСКИХ ИНФРАСТРУКТУР

Павел Андреевич Ломов¹, Андрей Владимирович Маслобоев², Андрей Григорьевич Олейник³

^{1, 2, 3}*Институт информатики и математического моделирования имени В. А. Путилова
Кольского научного центра Российской академии наук, Апатиты, Россия*

¹*p.lomov@ksc.ru, <https://orcid.org/0000-0002-0924-0188>*

²*a.masloboev@ksc.ru, <https://orcid.org/0000-0002-1231-9225>*

³*a.oleynik@ksc.ru, <https://orcid.org/0000-0002-7612-5999>*

Аннотация

В работе представлен вариант решения задачи формирования онтологии поддержки цикла управления жизнеспособностью критических инфраструктур. Онтология формируется на основе разработанной формализованной концептуальной модели процесса управления. В концептуальной модели представляются множества информационных сущностей и отношений, задействованных в различных фазах жизненного цикла процесса управления, а также необходимые атрибуты, характеризующие сущности и отношения. Концептуальное описание позволяет сформировать соответствующую систему онтологических паттернов содержания. Использование этих паттернов существенно упрощает создание прикладных онтологий для конкретных ситуаций, возникающих при функционировании критических инфраструктур. Для генерации паттернов реализуется двухэтапная процедура с использованием больших языковых моделей.

Ключевые слова:

критическая инфраструктура, жизнеспособность, цикл управление, онтология, онтологические паттерны содержания, большая языковая модель

Благодарности:

работа выполнена в рамках научно-исследовательской работы «Методы и информационные технологии мониторинга и управления региональными критическими инфраструктурами Арктической зоны Российской Федерации» (проект № FMEZ-2025-0054).

Для цитирования:

Ломов П. А., Маслобоев А. В., Олейник А. Г. Подход к формированию онтологии цикла управления жизнеспособностью критических инфраструктур // Труды Кольского научного центра РАН. Серия: Технические науки. 2025. Т. 16, № 3. С. 22–34. doi:10.37614/2949-1215.2025.16.3.002.

Original article

AN APPROACH TO ONTOLOGY DESIGN OF THE CRITICAL INFRASTRUCTURE RESILIENCE MANAGEMENT CYCLE

Pavel A. Lomov¹, Andrey V. Masloboev², Andrey G. Oleynik³

^{1, 2, 3}*Putilov Institute for Informatics and Mathematical Modeling of the Kola Science Centre of the Russian Academy of Sciences, Apatity, Russia*

¹*p.lomov@ksc.ru, <https://orcid.org/0000-0002-0924-0188>*

²*a.masloboev@ksc.ru, <https://orcid.org/0000-0002-1231-9225>*

³*a.oleynik@ksc.ru, <https://orcid.org/0000-0002-7612-5999>*

Abstract

A solution for the task of ontology forming for supporting the critical infrastructure resilience management cycle is presented in the article. The ontology is formed on the basis of the developed formalized conceptual model of the management process. The conceptual model represents sets of information entities and relationships involved in various phases of the life cycle of the management process, as well as the necessary attributes characterizing the entities and relationships. The conceptual description allows to form a corresponding system of Content Ontology Design Patterns. The use of these patterns significantly simplifies the creation of applied ontologies for specific situations arising during the operation of critical infrastructures. A two-step procedure using large language models is implemented to generate patterns.

Keywords:

critical infrastructure, resilience, management cycle, ontology, content ontology design patterns, large language models

Acknowledgments:

The study was carried out within the framework of the research project “Methods and information technologies for monitoring and managing regional critical infrastructures of the Arctic zone of the Russian Federation” (project No. FMEZ-2025-0054).

For citation:

Lomov P. A., Masloboev A. V., Oleynik A. G. An approach to ontology design of the critical infrastructure resilience management cycle. *Trudy Kol'skogo nauchnogo centra RAN. Seriya: Tekhnicheskie nauki* [Transactions of the Kola Science Centre of RAS. Series: Engineering Sciences], 2025, Vol. 16, No. 3, pp. 22–34. doi:10.37614/2949-1215.2025.16.3.002.

Введение

В состав Российской Федерации входят довольно обширные территории, которые характеризуются неблагоприятными климатическими условиями, высокими экологическими рисками, труднодоступностью и, как следствие, ограниченным развитием инфраструктурной обеспеченности. Поэтому на этих территориях особого внимания требуют надежность, безопасность и эффективность функционирования существующей инфраструктуры обеспечения различных видов деятельности, а также рациональное планирование новых инфраструктурных проектов. Обладающие определенной спецификой задачи обеспечения устойчивого функционирования инфраструктурных объектов не менее важны и для густонаселенных территорий с развитой промышленностью. Использование современных методов и информационных технологий мониторинга и управления инфраструктурами различного типа позволяет повысить их устойчивость к различным угрозам, быстро реагировать на аварийные ситуации и оптимизировать процессы управления.

Проводимые исследования направлены на разработку методов и средств информационной поддержки управления жизнеспособностью критических инфраструктур (КИ). КИ определяется как многокомпонентная распределенная система, состоящая из множества взаимозависимых подсистем, нарушение работоспособности хотя бы одной из которых может привести к существенному ухудшению безопасности жизнедеятельности населения (в широком смысле) и оказать значительное влияние на функционирование других подсистем и на жизнеспособность КИ в целом [1]. Под жизнеспособностью понимается защитное свойство системы [2], подвергающейся негативным воздействиям, позволяющее ей сопротивляться, поглощать, приспосабливаться и восстанавливаться после последствий воздействия, в том числе путем сохранения и восстановления ее основных структур и функций.

Управление жизнеспособностью КИ представляет собой многоэтапный и многофункциональный процесс, в общем случае включающий в себя такие функции управления, как целеполагание, стратегическое планирование, оперативное реагирование, а также функции контроля, учета, мониторинга, прогнозирования и координации. На каждом этапе этого процесса для реализации соответствующих функций управления необходимы ресурсы, силы и средства, адекватные оперативному контексту ситуации и решаемым задачам ситуационного управления.

Базовый цикл управления жизнеспособностью КИ включает последовательность взаимосвязанных фаз ситуационного управления устойчивым функционированием системы в условиях возникновения, развития и воздействия деструктивных событий различной природы и масштаба. Как правило, такой цикл включает в себя фазу идентификации, восприятия и предупреждения потенциальных угроз и опасностей, соответствующую режиму готовности (режим нормального функционирования системы), фазу поглощения негативного воздействия угроз и реагирования на возмущения и фазу восстановления работоспособности компонентов системы после сбоя. После фазы восстановления наступает фаза адаптации к новым условиям функционирования, в рамках которой система трансформируется и накапливает опыт противодействия угрозам, порожденным деструктивным воздействием на систему, тем самым повышая свою устойчивость (жизнеспособность) к подобного рода событиям в будущем.

Формализация задачи информационной поддержки управления жизнеспособностью

Для реализации средств информационно-аналитической поддержки управления жизнеспособностью КИ разрабатываются варианты формальных моделей, учитывающие специфику различных задач управления. Систематизация и примеры таких моделей подробно рассматриваются в работе [2].

В настоящей статье разработка варианта формального представления модели жизненного цикла управления жизнеспособностью системы опиралась на опыт использования технологии концептуального моделирования, основанной на функционально-целевом подходе [3], наработок в сфере ситуационного управления [4], онтологического моделирования [5; 6], методов анализа рисков нарушения безопасности КИ [7], кризисного управления [8], модельного инструментария и средств обеспечения жизнеспособности сложных систем [9].

В первую очередь были определены основные объекты (сущности), которые должны быть представлены в модели. К сущностям верхнего уровня относятся: объект управления, субъект управления и воздействия, несущие угрозы функционированию объекта управления. Каждая из этих сущностей имеет некоторую структуру и обладает определенными характеристиками. В рамках задач поддержки управленческих решений не всегда есть необходимость полностью рассматривать все элементы структуры и характеристики этих сущностей. Целесообразность учета компонентов структуры и характеристик, как правило, будет зависеть от конкретной ситуации, имеющей место на объекте управления. Ситуация, в свою очередь, определяется состоянием объекта управления и влиянием на его состояние некоторого деструктивного воздействия, нарушающего нормальное функционирование объекта управления. Возникновение нарушения порождает проблему. Устранение проблемы является целью субъекта управления и должно обеспечиваться им путем решения комплекса задач. Для решения задач субъект управления реализует определенные функции и использует необходимые ресурсы. В общем случае задачи и функции их решения распределяются между структурными компонентами субъекта управления, на основе которых формируется организационная структура управления решением задач устранения проблемы. Таким образом, организационная структура включает множество агентов управления, реализующих необходимые функции. В результате анализа наличия ресурсов и времени, необходимого для реализации каждой функции, формируется функциональная структура устранения проблемы. По аналогии с концептуальными моделями [3; 10] последовательность выполнения функций (отношения следования функций) может определяться двумя способами:

1) явно — задаются директивно, когда на основе нормативных документов или собственных предпочтений субъект управления явно указывает, что некоторая функция управления cf_k может выполняться только после завершения выполнения функции cf_i ;

2) неявно — определяются по потоку ресурсов — функция управления cf_k должна выполняться после завершения выполнения функции cf_i , если необходимые для выполнения cf_k ресурсы генерируются в результате выполнения функции cf_i .

Специальные процедуры автоматизированного анализа отношений следования на функциях, включенных в модель организационной структуры, позволяют проверить непротиворечивость явных и неявных отношений следования и генерировать функциональную модель управления. При определении порядка следования функций также учитывается время, необходимое на их выполнение.

В работе [11] впервые предложена формализованная структура модели субъекта управления, основанная на применении процессного подхода к анализу и моделированию ситуационного управления жизнеспособностью КИ и отличающаяся полнотой формального представления задач и функций обеспечения жизнеспособности и связанных с этими задачами процессов ситуационного управления.

На основании приведенных выше рассуждений в модели системы управления жизнеспособностью представлены следующие компоненты и множества:

$CF^l, l = \overline{1, n}$ — множество фаз жизненного цикла управления жизнеспособностью (n — количество фаз цикла);

$\{cf_x\}$ — множество функций управления, элементы которого соотносятся с определенными фазами, т. е. каждой фазе соответствует некоторое подмножество из m функций обеспечения жизнеспособности $cf_{ij}, i = \overline{1, n}, j = \overline{1, m}$. Каждая функция предполагает использование определенных ресурсов r_{ij} и времени t_{ij} , необходимых для ее реализации; m — количество функций, реализуемых в рамках i -й фазы.

cf_{i0} и cf_{ie} — начальная и конечная функции цикла, обеспечивающие подготовку входных ресурсов (например, «Мониторинг» или «Восприятие рисков») и получение результатов реализации цикла управления («Обучение» или «Оценка нового состояния») соответственно;

$MA = \{ma_x\}$ — множество агентов управления жизнеспособностью, выполняющих определенные функции в рамках этапов жизненного цикла жизнеспособности;

$R = \{r_q\}$, $q = \overline{1, l}$ — множество разнотипных ресурсов, необходимых для реализации функций управления cf_{ij} на этапах жизненного цикла управления жизнеспособностью. Для каждого ресурса в качестве обязательных атрибутов указывается его объем (Vol), тип (Typ) и стоимость (C).

На множестве MA могут быть заданы отношения подчиненности, обеспечивающие отражение иерархической структуры субъекта управления. Данные отношения могут быть полезны при разработке и использовании автоматизированных процедур координации действий агентов управления различных уровней в процессе решения задач, обусловленных конкретной ситуацией. Подобные процедуры представлены, например, в [12].

Каждому элементу множества агентов управления MA должно быть сопоставлено некоторое, реализуемое им, непустое подмножество функций управления $\{cf_{ij}\}_{mai}$. Следует отметить, что в общем случае одна и та же функция может реализовываться разными агентами управления, т. е. возможны ситуации, когда $\{cf_{ij}\}_{mai} \cap \{cf_{kx}\}_{mak} \neq \emptyset$.

Выше уже было указано, что между функциями, реализуемыми агентами управления, существуют отношения следования, которые могут определяться двумя способами - директивно, либо по потоку ресурсов. Явно в модели целесообразно представлять только отношения следования, задаваемые директивно: $CR = \{cr_{kj}\} = \{\langle cf_{ik}, cf_{ij} \rangle\}$, где кортеж $\langle cf_{ik}, cf_{ij} \rangle$ соответствует утверждению, что функция cf_{ik} может выполняться только после завершения выполнения функции cf_{ij} . При этом для корректного формирования модели управления должна быть предусмотрена процедура, проверяющая непротиворечивость отношений следования функций, определенных различными способами.

Между ресурсами и функциями определяются отношения использования и порождения ресурсов. Ресурсы, необходимые для выполнения функции (используемые функцией), относятся ко входным ресурсам данной функции и определяются в модели отношением $in(r_q, cf_{ij})$. Ресурсы, генерируемые в результате выполнения функции, относятся к выходным: $out(cf_{ij}, r_q)$. Аналогичным образом определяются входные и выходные ресурсы каждой фазы процесса управления. $IN^R(CF^i)$ — входные ресурсы, существующие (или доступные) к моменту начала фазы и необходимые для реализации выполняемых в рамках фазы функций. $OUT^R(CF^i)$ — выходные ресурсы, доступные по завершении фазы. Также для фазы имеет смысл рассматривать внутренние ресурсы $DOM^R(CF^i)$, которые не входят в $IN^R(CF^i)$, а генерируются функциями фазы. В зависимости от типов ресурсов указанные множества могут иметь непустое пересечение. Например, поступающие на вход фазы информационные ресурсы могут использоваться входящими в нее функциями, но передаваться на выход без изменений, а материальные ресурсы в процессе реализации фазы обязательно претерпевают изменения.

В связи с тем, что объект управления (критическая инфраструктура) обладает структурной и функциональной сложностью, попытки построения ее полной модели представляются малопродуктивными. Однако формирование адекватной модели фрагмента КИ, рассмотрение которого будет достаточно для поиска решения проблем, возникающих в конкретной ситуации, вполне возможно. Например, в работе [13] предложена информационная структура для информационной системы категорирования критически важных объектов (КВО), предусматривающая представление структуры и функций КВО, возможных угроз (деструктивных воздействий) и оценок последствий реализации угроз на компоненты и функции КВО. Аналогично в разрабатываемой модели должна быть реализована возможность представления фрагмента КИ, подвергающегося определенному воздействию в конкретной ситуации. Рассматриваемый фрагмент КИ может быть описан аналогично тому, как это реализовалось в системах концептуального моделирования [3; 10]. Для этого в модель включаются следующие множества:

$SC = \{sc_x\}$ — множество принимаемых в рассмотрение структурных компонентов КИ;

$F^{sc} = \{f_x^{sc}\}$ — множество принимаемых в рассмотрение функций и/или процессов, реализуемых элементами множества SC ;

$R^{Fsc} = \{r_x^{Fsc}\}$ — множества ресурсов, используемых и производимых элементами множества F^{sc} .

На множестве SC задаются отношения иерархии, определяющие структурную организацию принимаемых в рассмотрение компонентов КИ. Использование отношений иерархии типа «И» позволяет представить компонент КИ как композицию его составных частей. Использование типа иерархии «ИЛИ» дает возможность представить альтернативные варианты составных частей компонента.

Подобно описанным выше отношениям, задаваемым на множествах системы управления жизнеспособностью, задаются отношения и в модели фрагмента КИ. С каждым элементом множества SC связывается непустое подмножество множества F^{sc} , что определяет, какие функции реализует конкретный элемент множества SC . Равносильным является установка связей от элементов F^{sc} к элементам SC , показывающих, какими компонентами КИ может реализовываться конкретная функция. Также устанавливаются отношения «вход/выход» между элементами множеств F^{sc} и $R^{F^{sc}}$ и отношения следования на множестве F^{sc} , определяющие порядок их выполнения. Отношения на модели фрагмента КИ позволяют более детально определить угрозы или результаты деструктивного воздействия на КИ и идентифицировать порожденную проблему и комплекс задач, обеспечивающих ее устранение.

Проблемам классификации, моделирования, оценки рисков и предупреждения чрезвычайных ситуаций посвящено довольно много научных работ. В частности, издательством «МГФ Знание» выпускается серия книг «Безопасность России», в которых систематизируются результаты широкого спектра исследований и разработок в области анализа проблем риска и безопасности. Существует обширная нормативная база, регулирующая деятельность по предотвращению опасностей и рисков, ликвидации или снижению негативных последствий от чрезвычайных ситуаций различной природы. На основе имеющейся информации можно сформировать модель деструктивного воздействия на КИ, необходимую и достаточную для выявления порождаемой им в конкретной ситуации проблемы и определения комплекса задач по ее устранению.

Структура представления в модели деструктивных воздействий на КИ включает множества идентификаторов событий воздействия $DA = \{da_i\}$, типов воздействий $TA = \{ta_j\}$ и угроз/рисков, порождаемых воздействиями, $TR = \{tr_k\}$. Для каждого принимаемого в рассмотрение деструктивного воздействия da_i , кроме собственно идентификатора (наименования), задается время и место события. Путем установления связи da_i с элементами множества TA определяются типы порождаемых этим событием воздействий. На основе нормативных документов и других литературных источников (например, [14]) в системе информационно-аналитической поддержки управления может быть создан «справочник» типов воздействий, а связь ta_j с элементами множества TR определяет, какие риски возникают вследствие данного воздействия. Для связи между воздействием и риском ($\langle ta_j, tr_k \rangle$) задаются атрибуты, определяющие величину и вероятность риска в конкретной рассматриваемой ситуации.

Формирование модели для поддержки управления жизнеспособностью КИ в конкретной ситуации осуществляется путем задания соответствующих экземпляров элементов описанных выше множеств и текущих значений атрибутов этих элементов. Затем устанавливаются связи, определяющие воздействие элементов, характеризующих деструктивное событие, на элементы, характеризующие состав и функции компонентов КИ. Для этих связей могут быть заданы атрибуты, определяющие степень и вероятность воздействия. На основе анализа влияния текущего события на компоненты КИ определяется проблема и комплекс задач для ее устранения. Анализ может проводиться как экспертным путем, так и с использованием автоматизированных процедур, основанных на различных моделях оценки рисков. Затем для требующих решения задач выбираются «покрывающие» [10] функции управления cf_{ij} и акторы управления, которые могут реализовать эти функции ma_i . При необходимости на выбранном подмножестве функций задаются директивные отношения следования. Таким образом будут определены все элементы организационной структуры для решения возникшей в конкретной ситуации проблемы. Следует отметить, что при установлении связей между элементами описаний деструктивного воздействия, компонентов КИ и системы управления состав принимаемых к рассмотрению экземпляров элементов может изменяться. Это обусловлено как выявлением наличия неучтенных изначально элементов, так и определением избыточности начальных описаний.

Построение онтологической модели цикла управления жизнеспособностью

Для компьютерной реализации концептуальных моделей предметных областей и решаемых задач в последние десятилетия широко используются формальные онтологии [6; 15]. Созданы и развиваются программные системы построения и обработки онтологий [16; 17]. Однако построение концептуальной модели сложной системы или задачи «вручную» даже с использованием специальных редакторов онтологий — весьма трудоемкая процедура.

Существенно упрощает процесс построения онтологии использование онтологических паттернов, которые представляют собой детально описанные и проверенные на практике решения регулярно возникающих проблем онтологического моделирования [18; 19]. Они позволяют повторно использовать удачные проектные решения и обеспечивают единообразие структуры и семантики создаваемых онтологий. В зависимости от решаемых задач различают несколько типов онтологических паттернов [20].

В настоящей работе, как и в [21], используются онтологические паттерны содержания (Content Ontology Design Patterns, CDP), так как они непосредственно определяют структуру понятийной системы разрабатываемой онтологии. Как было отмечено ранее, они представляют собой фрагменты онтологий, предназначенные для повторного использования при моделировании распространенных ситуаций, возникающих между объектами, процессами и явлениями в различных предметных областях, таких как, например, участие объектов в событиях, отношения часть — целое, последовательности действий, причинно-следственные зависимости и др.

Фактически паттерны содержания можно рассматривать как структурированные фрагменты онтологий верхнего уровня. Они содержат общие классы (Агент, Последовательность), свойства (идентификатор, имя) и отношения (зависит от, имеет часть), которые могут быть конкретизированы. После такой специализации содержащихся в них элементов и добавления новых элементов CDP становятся неотъемлемыми составными частями прикладных онтологий.

Использование CDP в процессе проектирования онтологий способствует унификации их моделей и согласованности. Каждый паттерн содержания отражает определенную точку зрения на понятие (conceptual viewpoint), которая должна коррелировать с задачами моделирования конкретной предметной области или задачи. Для выбора наиболее подходящего паттерна применяются так называемые компетентностные вопросы (Competency Questions, CQ) [20]. Эти вопросы определяют, какие сведения можно получить из онтологии, если в ее структуре используется данный паттерн. Таким образом, набор CQ служит инструментом для проверки релевантности CDP целям моделирования и помогает разработчику выбрать паттерн, представляющий подходящую схему описания знаний.

Сами паттерны содержания целесообразно генерировать с использованием больших языковых моделей (Large Language Models, LLM). В настоящее время существуют работы, посвященные применению языковых моделей [22; 23] в онтологическом моделировании, однако они преимущественно ориентированы на создание и/или наполнение полноценных онтологий. Такой подход требует значительных вычислительных и когнитивных ресурсов, а также сложной верификации.

В отличие от этого, генерация онтологических паттернов содержания представляет собой намного более простую задачу, так как паттерн концептуально проще полной онтологии. Он описывает решение одной конкретной задачи моделирования и, как правило, включает ограниченное число сущностей и отношений. Благодаря этому при генерации описание требуемого паттерна можно задать в виде краткой инструкции (промпта) нескольких предложений, содержащих спецификацию типовой ситуации, которую необходимо отразить в онтологии.

Для повышения контроля над процессом генерации, а также снижения сложности инструкций для языковой модели и тем самым уменьшения вероятности получения синтаксических ошибок и/или так называемых галлюцинаций имеет смысл проводить генерацию в два этапа.

На первом этапе выполняется формирование набора функций на выбранном языке программирования, которые принимают на вход простые структуры данных — массивы (списки), ассоциативные массивы (словари), вложенные структуры, которые содержат естественно языковые имена понятий, отношений и атрибутов — и текущую онтологию. Каждая функция реализует определенный тип модификации онтологической модели (например, добавление класса, отношения, атрибута, ограничения). На выходе функции возвращают модифицированную онтологию, включающую новые элементы, описанные во входных данных. Таким образом, язык программирования выступает в роли прослойки между LLM и онтологическим форматом, обеспечивая контроль корректности синтаксиса и структуры создаваемых фрагментов.

В качестве примера рассмотрим промпт для генерации функции, добавляющей список понятий в онтологию:

Напиши на Python функцию для наполнения онтологии OWL с использованием библиотеки owlready2.

Функция должна принимать следующие аргументы:

** объект онтологии (Ontology) из библиотеки owlready2;*

** словарь, в котором ключ — имя понятия, а значение — его определение.*

Функция должна добавлять понятия в онтологию как классы, а их определения — как аннотации
rdfs:comment.

Добавь подробный комментарий (docstring), описывающий назначение функции и её аргументы.

Результатом генерации с помощью языковой модели GigaChat будет следующая функция:

```
from owlready2 import *
```

```
def populate_ontology_with_classes_and_comments(onto: Ontology, concepts_dict: dict):
```

```
    """
```

Функция наполняет онтологию новыми классами и добавляет аннотации к ним.

Параметры:

onto (owlready2.Ontology): Объект онтологии, куда будут добавляться новые классы.

*concepts_dict (dict): Словарь, где ключи - имена понятий (названия классов),
а значения - их описания (аннотации).*

Действия:

1. Для каждого элемента словаря создается новый класс с именем ключа.

2. Классу добавляется аннотация 'rdfs:comment' с текстом, указанным в значении словаря.

```
    """
```

```
    for concept_name, definition in concepts_dict.items():
```

```
        # Создаем новый класс в онтологии с заданным именем
```

```
        new_class = types.new_class(concept_name, (Thing,), ontology=onto)
```

```
        # Добавляем аннотацию rdfs:comment с описанием класса
```

```
        new_class.comment.append(definition)
```

На втором этапе формируется инструкция для языковой модели, включающая запрос на генерацию набора элементов будущего паттерна — классов, свойств, отношений и ограничений — представленных в виде структур данных, принимаемых функциями, созданными на первом этапе. В этом случае LLM не формирует онтологию напрямую, а лишь порождает описание паттерна в виде структурированных данных, которые используются для получения окончательного результата с помощью функций на языке программирования.

Например, с помощью такого промпта можно извлечь из текста — описания паттерна понятия, которые далее можно поместить в его онтологию с помощью ранее приведенной функции:

Из представленного текста необходимо извлечь информацию для построения OWL-онтологии.

Построй структуру онтологии, соответствующую содержанию текста, и представь результат в виде трёх Python-структур:

concepts — словарь, где — имя понятия, значение — определение понятия.

object_properties — список кортежей, где каждый кортеж имеет вид: (subject_concept, property_name, range_concept, описание_свойства) и отражает отношения между понятиями (object properties).

datatype_properties — список кортежей, где каждый кортеж имеет вид: (subject_concept, property_name, тип значения, описание_свойства) и отражает атрибуты (data properties) понятий.

При указании типов значений используй корректные OWL/XSD-обозначения:

xsd:string, xsd:integer, xsd:float, xsd:boolean, xsd:dateTime.

В результате будет получен список понятий в виде python-словаря:

```
concepts = {
```

```
"ResilienceCycle": "Формальная модель управления устойчивостью системы, включающая взаимосвязанные фазы, функции, ресурсы и агентов.",
```

```
"Phase": "Фаза жизненного цикла устойчивости: anticipation, absorption или adaptation.",
```

```
"ControlFunction": "Управляющая функция, выполняемая в определённой фазе цикла для поддержания устойчивости.", ... }
```

Такой подход позволяет четко разделить фазу концептуализации знаний (осуществляемую LLM и разработчиком онтологии) и фазу формальной спецификации (выполняемую средствами программного представления и сериализации онтологий). Это обеспечивает: 1) синтаксическую корректность результата, поскольку ответственность за генерацию формальных структур возлагается на проверенные программные библиотеки для работы с онтологическими форматами (например, OWL API, RDFLib, Owlready2, rdflib.js); 2) снижение вероятности семантических и логических ошибок, так как языковая модель оперирует ограниченным числом понятий и отношений, представленных в небольшом текстовом описании паттерна; 3) получение разработчиком возможности легко проверять и уточнять результат, поскольку результат концептуализации, сгенерированный языковой моделью, будет представлен в виде относительно простых структур языка программирования.

Разработку онтологии с использованием приведенного подхода удобнее осуществлять в так называемых электронных блокнотах (Computational Notebooks), которые можно запускать в программных средах Google Colab, Jupyter Notebook и Apache Zeppelin. Такие блокноты позволяют комбинировать форматированные текстовые заметки с блоками программного кода, а также легко запускать последние в необходимом разработчику порядке.

С помощью такого подхода на основе соответствующих фрагментов приведенной выше формализации могут быть построены паттерны, описывающие ключевые аспекты системы управления жизнеспособностью. Для примера ниже приведены схемы трех сгенерированных паттернов (рис. 1–3).



Рис. 1. Онтологический паттерн «Объект управления»

Паттерн «Объект управления» (см. рис. 1) определяет структуру онтологического описания объекта (системы), относительно которого осуществляется решение задач обеспечения жизнеспособности. Наряду с функциями, или «услугами», которые реализует данный объект, и их характеристиками, в паттерн включаются «потребители» услуг (класс "SystemFunctionConsumer"). Это дает возможность при использовании данного паттерна для формирования онтологической модели конкретной ситуации достаточно полно представить «сферу влияния» [13] данного объекта на другие объекты и системы, зависящие от его функционирования, что, в свою очередь, способствует более адекватной оценке критичности как самого объекта управления, так и сложившейся в результате деструктивного воздействия ситуации.

Паттерн описания деструктивного события (см. рис. 2) предусматривает указание места его возникновения, или проявления (класс *"EventPosition"*), а также то, как данное событие проявляется (класс *"EventManifestation"*). Отметим, что в общем случае одно событие может иметь несколько проявлений, оказывающих разное влияние на функциональность объекта управления. Для каждого проявления задается сила его проявления в данном событии.

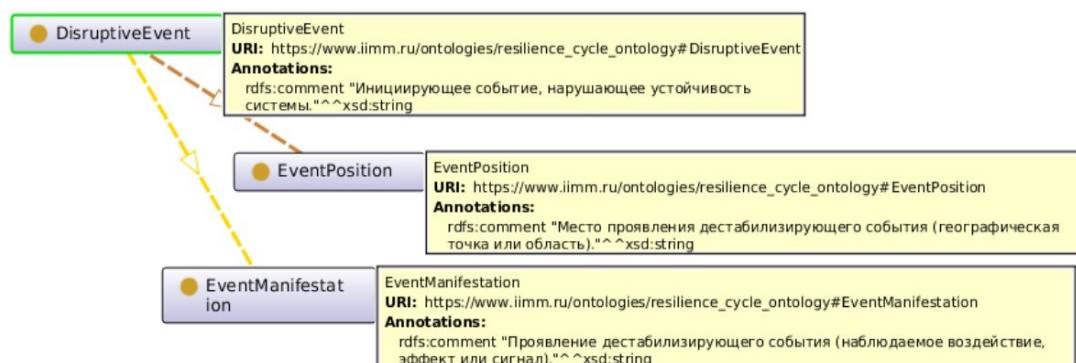


Рис. 2. Онтологический паттерн «Деструктивное событие»

Паттерн «Факт воздействия» (см. рис. 3) позволяет сформировать онтологическое описание воздействия конкретного деструктивного события на объект управления путем указания, какие проявления события, в какой степени влияют на определенные функции объекта. При этом существует возможность указать значение силы влияния, которое может быть отличным от данной характеристики проявления в описании деструктивного события.

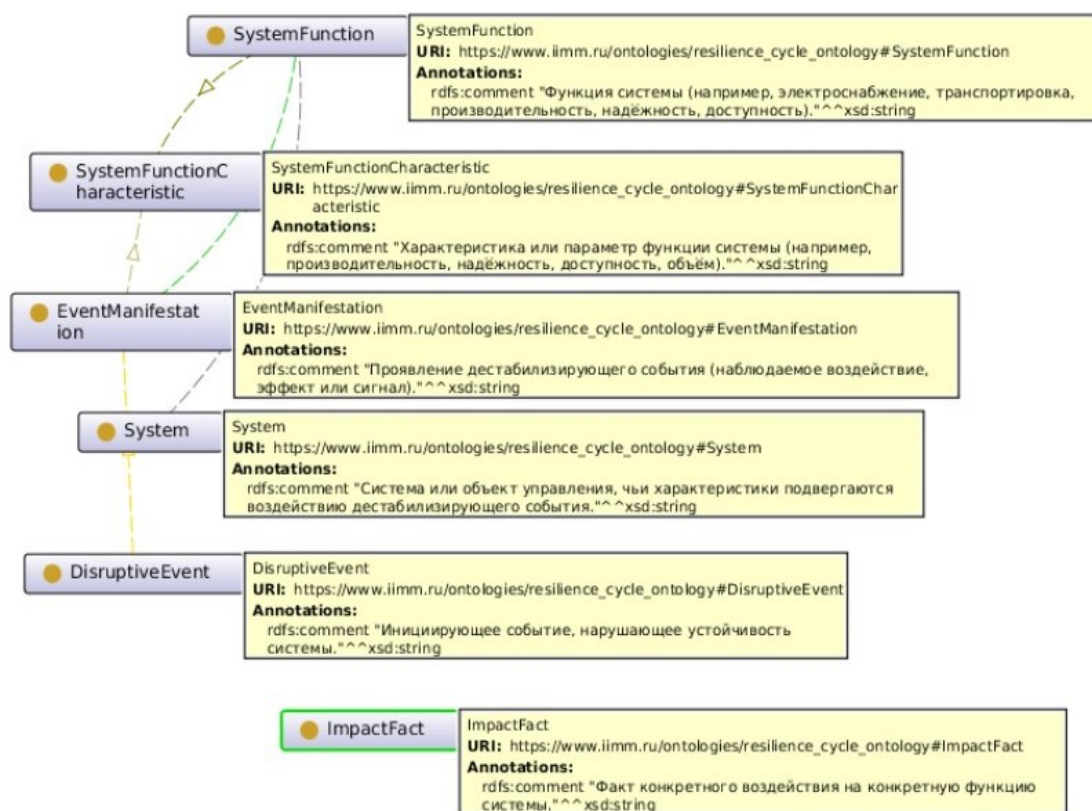


Рис. 3. Онтологический паттерн «Факт воздействия»

Представленные в качестве примеров паттерны можно условно отнести к категории элементарных. Они включают небольшое число компонентов (классов и отношений) и просты в использовании.

Путем комбинирования элементарных паттернов формируются композитные паттерны. Они представляют структуру взаимодействия элементарных паттернов, а также сложные объекты и взаимодействия предметной области.

Заключение

Предложенная формализованная структура концептуальной модели поддержки управления жизнеспособностью критических инфраструктур определяет состав классов информационных сущностей и связей между ними, задействованных в реализации всех фаз жизненного цикла процесса управления. Она включает три основных блока, представляющих систему управления жизнеспособностью, собственно объект управления, в качестве которого рассматривается критическая инфраструктура, и различные события, или процессы, оказывающие деструктивное воздействие на объект управления.

В качестве компьютерной реализации модели предложено создать онтологию на основе онтологических паттернов содержания (CDP). Для генерации паттернов использованы возможности больших языковых моделей (LLM), которые генерировали программный код паттерна в соответствии с задаваемой текстовой инструкцией, содержащей спецификацию концептуальной структуры, требующей представления в онтологии. Генерация осуществляется в два этапа, что повышает контроль над процессом генерации и снижает вероятность получения синтаксических ошибок и/или так называемых галлюцинаций в генерируемых LLM кодах.

Полный комплекс паттернов, содержащих представление всех компонентов формальной концептуальной модели процесса поддержки управления жизнеспособностью КИ, предоставит пользователю необходимый набор инструментов для предметного описания подлежащих анализу проблемных ситуаций. Онтологическая модель для конкретной ситуации формируется путем задания экземпляров представленных в паттернах сущностей, значений атрибутов этих экземпляров и определенных шаблонами связей между ними. Процедуры вывода на полученной онтологии обеспечивают формирование организационной и функциональной модели управления решением задач по устранению представленной в онтологии проблемы. В результате будет получена онтология, являющаяся экземпляром онтологии задачи управлением поддержанием жизнеспособности конкретной критической инфраструктуры при определенном деструктивном воздействии.

Несмотря на то, что в общем случае ситуации нарушения нормального функционирования критических инфраструктур имеют свои особенности, сохранение уже построенных онтологических моделей полезно, так как их фрагменты в различном объеме могут быть использованы при построении онтологических моделей управления поддержанием жизнеспособности КИ и в других ситуациях. Например, могут рассматриваться аналогичные деструктивные воздействия, иметь место нарушение подобных компонентов КИ или в решении возникших проблем могут участвовать такие же акторы управления. Учитывая это, целесообразно обеспечить повторное использование не только CDP-паттернов, но и разработанных на их основе фрагментов прикладных онтологий. Для этого и каталог CDP-паттернов, и фрагменты онтологий должны быть размещены в публичном репозитории с регламентированным доступом.

Список источников

1. Маслобоев А. В., Цыгичко В. Н. Адаптация и развитие риск-ориентированной методологии обеспечения безопасности критически важных объектов к управлению жизнеспособностью критических инфраструктур // Надежность и качество сложных систем. 2024. № 4. С. 140–159.
2. Маслобоев А. В. Формальные модели жизнеспособности региональных критических инфраструктур // Труды Института системного анализа РАН. 2022. Т. 72, № 3. С. 59–80.
3. Олейник А. Г., Путилов В. А. Развитие технологии концептуального моделирования, основанной на функционально-целевом подходе // История науки и техники. 2014. № 1. С. 37–52.

4. Фридман А. Я., Олейник А. Г. Методы и средства поддержки принятия решений по обеспечению устойчивого функционирования промышленно-природных комплексов в Арктической зоне РФ // История науки и техники. 2019. № 4. С. 26–34.
5. Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В., Кашевник А. М. Контекстно-управляемая поддержка принятия решений в распределенной информационной среде // Информационные технологии и вычислительные системы. 2009. № 1. С. 38–48.
6. Husáková M., Bureš V. Formal Ontologies in Information Systems Development: A Systematic Review // Information. 2020. Vol. 11, Iss. 2. Article no.: 0066.
7. Цыгичко В. Н., Черешкин Д. С., Смолян Г. Л. Безопасность критических инфраструктур. М.: УРСС, 2019. 200 с.
8. Pursiainen C. The Crisis Management Cycle. UK, London: Routledge, 2017. 194 p.
9. Critical Infrastructure Security and Resilience: Theories, Methods, Tools and Technologies // Advanced Sciences and Technologies for Security Applications / ed. by D. Gritzalis, M. Theocharidou, G. Stergiopoulos. Springer Cham, Springer Nature Switzerland AG, 2019. 313 p.
10. Емельянов С. В., Олейник А. Г., Попков Ю. С., Путилов В. А. Информационные технологии регионального управления. М.: Едиториал УРСС, 2004. 400 с.
11. Masloboev A. V. A ternary conceptual framework of critical infrastructure resilience management cycle // Reliability and Quality of Complex Systems. 2025. no. 3. P. 109–125.
12. Fridman A. Planning and Coordination in Hierarchies of Intelligent Dynamic Systems // TELKOMNIKA. December 2016. Vol. 14, No. 4. P. 1408–1416.
13. Яковлев С. Ю., Шемякин А. С., Олейник А. Г. Регулирование техногенно-экологической безопасности критически важных объектов инфраструктуры: обновление нормативной базы // Труды Кольского научного центра РАН. Серия: Технические науки. 2022. Т. 13, № 2. С. 93–102.
14. Анализ риска и проблем безопасности: в 4 ч. // 4.1. Основы анализа и регулирования безопасности; науч. руковод. К. В. Фролов. М.: МГФ «Знание», 2006. 640 с.
15. Deckers R., Lago P. Systematic Literature Review of Domain-Oriented Specification Techniques // Journal of Systems and Software. 2022. Vol. 192. Article no.: 111415.
16. Gehrke B., Braun M., Schenk P., West R., Michie S., Hastings J. OntoSpreadEd: Developing Ontologies with a Dedicated Online Template Spreadsheet Editor // Wellcome Open Research. 2025. Vol. 10. Article no.: 360.
17. Mateiu P., Groza A. Ontology Engineering with Large Language Models // 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). 2023. P. 226–229.
18. Gangemi A. Ontology Design Patterns for Semantic Web Content // The Semantic Web—ISWC 2005. Lecture Notes in Computer Science. 2005. Vol. 3729. P. 262–276.
19. Presutti V., Gangemi A., David S., de Cea G., Suárez-Figueroa M. C., Montiel-Ponsoda E., Poveda M. Ontology Design Patterns for Ontology Reuse and Re-engineering // Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008). 2008. P. 128–142.
20. Blomqvist E., Presutti V., Daga E., Gangemi A. Experimenting with Ontology Design Patterns in an Ontology Engineering Context // Proceedings of the 5th Workshop on Ontology and Semantic Web Patterns (WOP 2014). CEUR Workshop Proceedings. 2014. Vol. 1302. 83 p.
21. Олейник А. Г., Ломов П. А. Разработка онтологии интегрированного пространства знаний // Онтология проектирования. 2016. Т. 6, № 4 (22). С. 465–474.
22. Lo A., Jiang A. Q., Li W., Jamnik M. End-to-End Ontology Learning with Large Language Models // Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024). 2024. Article No.: 2767. P. 87184–87225.
23. Lippolis A. S., Saeedizade M. J., Keskisärkkä R., Zuppiroli S., Ceriani M., Gangemi A., Blomqvist E., Nuzzolese A. G. Ontology Generation using Large Language Models // Proceedings of the 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1–5, 2025. 2025. Part I. P. 321–341.

References

1. Masloboev A. V., Tsygichko V. N. Adaptatsiya i razvitie risk-orientirovannoy metodologii obespecheniya bezopasnosti kriticheskikh vazhnykh ob"ektov k upravleniyu zhiznesposobnost'yu kriticheskikh infrastruktur [Adaptation and development of a risk-based methodology for ensuring the security of critical facilities for managing the resilience of critical infrastructures]. *Nadezhnost' i kachestvo slozhnykh sistem* [Reliability and Quality of Complex Systems], 2024, no. 4, pp. 140–159. (In Russ.).

2. Masloboev A. V. Formal'nye modeli zhiznesposobnosti regional'nykh kriticheskikh infrastruktur [Formal models of the resilience of regional critical infrastructures]. *Trudy Instituta sistemnogo analiza RAN* [Proceedings of the Institute of Systems Analysis of the Russian Academy of Sciences], 2022, Vol. 72, no. 3, pp. 59–80. (In Russ.).
3. Oleynik A. G., Putilov V. A. Razvitie tekhnologii kontseptual'nogo modelirovaniya, osnovannoy na funktsional'no-tselevom podkhoda [Development of conceptual modeling technology based on a functional-target approach]. *Istoriya nauki i tekhniki* [History of Science and Technology], 2014, no. 1, pp. 37–52. (In Russ.).
4. Fridman A. Ya., Oleynik A. G. Metody i sredstva podderzhki prinyatiya resheniy po obespecheniyu ustoychivogo funktsionirovaniya promyshlenno-prirodnnykh kompleksov v Arkticheskoy zone RF [Methods and means of supporting decision-making to ensure sustainable functioning of industrial and natural complexes in the Arctic zone of the Russian Federation]. *Istoriya nauki i tekhniki* [History of Science and Technology], 2019, no. 4, pp. 26–34. (In Russ.).
5. Smirnov A. V., Pashkin M. P., Shilov N. G., Levashova T. V., Kashevnik A. M. Kontekstno-upravlyaemaya podderzhka prinyatiya resheniy v raspredelennoy informatsionnoy srede [Context-driven decision support in a distributed information environment]. *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information technology and computing systems], 2009, no. 1, pp. 38–48. (In Russ.).
6. Husáková M., Bureš V. Formal Ontologies in Information Systems Development: A Systematic Review. *Information*, 2020, Vol. 11, Iss. 2, Article no.: 0066.
7. Tsygichko V. N., Chereskin D. S., Smolyan G. L. *Bezopasnost' kriticheskikh infrastruktur* [Critical Infrastructure Security]. Moscow, URSS, 2019, 200 p. (In Russ.).
8. Pursiainen C. *The Crisis Management Cycle*. UK, London, Routledge, 2017, 194 p.
9. Critical Infrastructure Security and Resilience: Theories, Methods, Tools and Technologies. *Advanced Sciences and Technologies for Security Applications* / ed. by D. Gritzalis, M. Theodoridou, G. Stergiopoulos. Springer Cham, Springer Nature Switzerland AG, 2019, 313 p.
10. Emel'yanov S. V., Oleynik A. G., Popkov Yu. S., Putilov V. A. *Informatsionnye tekhnologii regional'nogo upravleniya* [Information technologies for regional management]. Moscow, Editorial URSS, 2004, 400 p. (In Russ.).
11. Masloboev A. V. A ternary conceptual framework of critical infrastructure resilience management cycle. *Reliability and Quality of Complex Systems*, 2025, no. 3, pp. 109–125.
12. Fridman A. Planning and Coordination in Hierarchies of Intelligent Dynamic Systems. *TELKOMNIKA*. December 2016, Vol. 14, no. 4, pp. 1408–1416.
13. Yakovlev S. Yu., Shemyakin A. S., Oleynik A. G. Regulirovanie tekhnogenno-ekologicheskoy bezopasnosti kriticheskikh vazhnykh ob"ektov infrastruktury: obnovenie normativnoy bazy [Regulation of technogenic and environmental safety of critical infrastructure facilities: updating the regulatory framework]. *Trudy Kol'skogo nauchnogo tsentra RAN. Seriya: Tekhnicheskie nauki* [Proceedings of the Kola Science Center of the Russian Academy of Sciences. Series: Technical Sciences], 2022, Vol. 13, no. 2, pp. 93–102. (In Russ.).
14. *Analiz riska i problem bezopasnosti: v 4 ch. 4.1. Osnovy analiza i regulirovaniya bezopasnosti* [Analysis of Risks and Safety Issues. In 4 Parts. 4.1. Fundamentals of Safety Analysis and Regulation]. Moscow, MGF “Znanie”, 2006, 640 p. (In Russ.).
15. Deckers R., Lago P. Systematic Literature Review of Domain-Oriented Specification Techniques. *Journal of Systems and Software*, 2022, Vol. 192, Article no.: 111415.
16. Gehrke B., Braun M., Schenk P., West R., Michie S., Hastings J. OntoSpreadEd: Developing Ontologies with a Dedicated Online Template Spreadsheet Editor. *Wellcome Open Research*, 2025, Vol. 10. Article no.: 360.
17. Mateiu P., Groza A. Ontology Engineering with Large Language Models. *25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2023, pp. 226–229.
18. Gangemi A. Ontology Design Patterns for Semantic Web Content. *The Semantic Web—ISWC 2005. Lecture Notes in Computer Science*, 2005, Vol. 3729, pp. 262–276.
19. Presutti V., Gangemi A., David S., de Cea G., Suárez-Figueroa M. C., Montiel-Ponsoda E., Poveda M. Ontology Design Patterns for Ontology Reuse and Re-engineering. *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*, 2008, pp. 128–142.
20. Blomqvist E., Presutti V., Daga E., Gangemi A. Experimenting with Ontology Design Patterns in an Ontology Engineering Context. *Proceedings of the 5th Workshop on Ontology and Semantic Web Patterns (WOP 2014). CEUR Workshop Proceedings*, 2014, Vol. 1302, 83 p.
21. Oleynik A. G., Lomov P. A. Razrabotka ontologii integrirovannogo prostranstva znaniy [Development of an ontology for an integrated knowledge space]. *Ontologiya proektirovaniya* [Ontology of design], 2016, Vol. 6, no. 4 (22), pp. 465–474. (In Russ.).

22. Lo A., Jiang A. Q., Li W., Jamnik M. End-to-End Ontology Learning with Large Language Models. *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024, Article No.: 2767, pp. 87184–87225.
23. Lippolis A. S., Saeedizade M. J., Keskiärrkkä R., Zuppiroli S., Ceriani M., Gangemi A., Blomqvist E., Nuzzolese A. G. Ontology Generation using Large Language Models. *Proceedings of the 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1–5, 2025*, 2025, Part I, pp. 321–341.

Информация об авторах

П. А. Ломов — кандидат технических наук, старший научный сотрудник;
А. В. Маслобоев — доктор технических наук, ведущий научный сотрудник;
А. Г. Олейник — доктор технических наук, главный научный сотрудник.

Information about the authors

P. A. Lomov — Candidate of Science (Tech.), Senior Research Fellow;
A. V. Masloboev — Doctor of Science (Tech.), Leading Research Fellow;
A. G. Oleynik — Doctor of Science (Tech.), Chief Research Fellow.

Статья поступила в редакцию 15.09.2025; одобрена после рецензирования 20.09.2025; принята к публикации 08.10.2025.
The article was submitted 15.09.2025; approved after reviewing 20.09.2025; accepted for publication 08.10.2025.