

УДК 004.4

**Евсеева Юлия Игоревна,**

Пензенский государственный университет, кафедра "Системы автоматизированного проектирования", кандидат технических наук, доцент.

**Клочкова Полина Евгеньевна,**

Пензенский государственный университет, кафедра "Системы автоматизированного проектирования", студент, 2 курс

**ИНТЕГРАЦИОННЫЙ ПОДХОД К СОЗДАНИЮ  
САМОАДАПТИВНЫХ ПРОГРАММНЫХ СИСТЕМ: АРХИТЕКТУРА  
И МЕХАНИЗМЫ АДАПТАЦИИ**

**Аннотация.** В статье предлагается комплексный подход к разработке самоадаптивных программных систем, сочетающий формальную строгость динамических линеек программных продуктов (DSPL) и моделей характеристик с семантическими возможностями больших языковых моделей (LLM). Описывается архитектура на основе модифицированного цикла MAPЕ-K, обеспечивающая замкнутый цикл адаптации. Подход позволяет создавать системы, эффективно функционирующие в условиях высокой неопределённости.

**Ключевые слова:** самоадаптивное программное обеспечение, динамические линейки программных продуктов, модели характеристик, большие языковые модели, архитектура.

**Введение**

Современные программные системы функционируют в условиях высокой неопределенности и непрерывно изменяющейся среды выполнения.

Традиционные подходы, основанные на статической конфигурации программного обеспечения и predetermined сценариях адаптации, недостаточно эффективны. Ранее одним из авторов работы были созданы методы синтеза самоадаптивного программного обеспечения на основе динамических линеек программных продуктов (DSPL) и моделей характеристик [1-3], обеспечивающие строгий контроль конфигураций программы, но ограниченные в работе с семантически насыщенным контекстом выполнения.

Предлагаемый подход сочетает три ключевые технологии: динамические линейки программных продуктов (dynamic software product lines, DSPL) [4], модели характеристик [5] и большие языковые модели (large language model, LLM) [6]. Формальная строгость DSPL и моделей характеристик обеспечивает надёжность реконфигурации системы в процессе выполнения, а семантические возможности LLM позволяют работать со сложными неструктурированными данными.

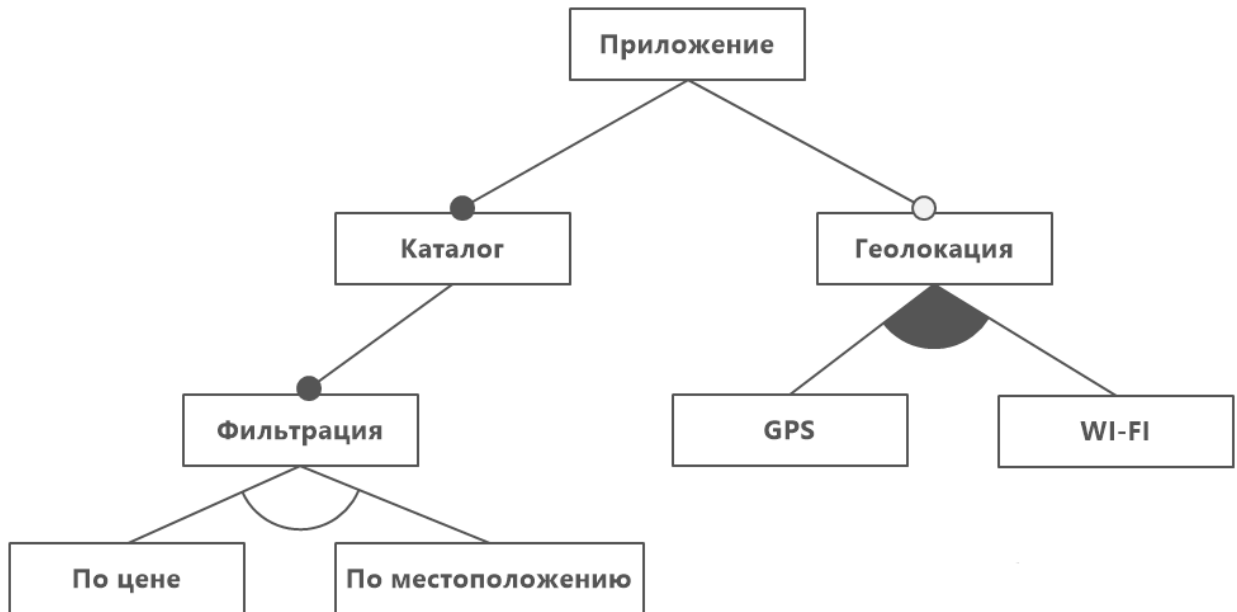
### **Технологические основы**

Основу предлагаемого подхода составляют три технологии, каждая из которых вносит свой специфический вклад в процесс самоадаптации.

Динамические линейки программных продуктов (DSPL) выступают архитектурной основой самоадаптивного программного обеспечения, обеспечивающей возможность динамической перестройки работающей системы в ответ на изменения контекста. Они поддерживают управление вариативностью, самодиагностику и самовосстановление, однако сами по себе не позволяют гибко интерпретировать семантически сложные входные данные.

Для формализации пространства возможных состояний системы используются модели характеристик, которые представляют вариативность через иерархии характеристик и логические ограничения, задавая тем самым

множество допустимых конфигураций. Под характеристикой здесь понимается отдельное функциональное или нефункциональное свойство



программной системы — например, наличие определённого модуля, способа аутентификации, уровня логирования или режима работы. Каждая характеристика может быть обязательной или опциональной, а связи между ними определяются ограничениями, такими как взаимное исключение или зависимость. Благодаря этому обеспечивается формальная строгость и проверяемость любых изменений. Пример модели характеристик представлен на рисунке 1.

Рисунок 1. Пример модели (диаграммы) характеристик

Семантическую гибкость вносит большая языковая модель (LLM), которая обеспечивает глубокий анализ контекста, извлечение смысла из неструктурированных данных (текстовые запросы, сообщения, логи), а также генерацию конфигураций и логический вывод на основе естественно-языковых описаний.

Таким образом, интеграция трёх технологий позволяет компенсировать ограничения каждой из них в отдельности: формальная строгость DSPL и моделей характеристик дополняется семантической гибкостью LLM.

## Архитектура системы и цикл самоадаптации

Архитектура строится вокруг модифицированной модели MARE-K (Monitor-Analyze-Plan-Execute-Knowledge) [7], включающей следующие компоненты:

1. Мониторинг (Monitor) реализуется через распределённую систему сенсоров, собирающих структурированные метрики и неструктурированную информацию (текстовые сообщения, логи, показания датчиков).
2. Анализ (Analyze) включает модуль семантического анализа на основе LLM, преобразующий сырые данные в формализованное представление контекста.
3. Планирование (Plan) осуществляет сопоставление контекста с пространством конфигураций программы, которое задается с помощью модели характеристик. LLM оценивает соответствие характеристик текущему контексту и генерирует оптимальную конфигурацию системы.
4. Исполнение (Execute) реализуется механизмами DSPL, обеспечивающими динамическую реконфигурацию с минимизацией сбоев и согласованностью состояний.
5. База знаний (Knowledge) аккумулирует успешные и неудачные сценарии адаптации, дополняя модель характеристик новыми ограничениями и предоставляя данные для дальнейшего обучения LLM.

Цикл самоадаптации функционирует следующим образом: система непрерывно собирает данные мониторинга; LLM преобразует неструктурированную информацию в формализованный контекст; на основе этого контекста формируется целевая конфигурация, максимизирующая полезность системы; план проверяется на соответствие ограничениям модели

характеристик, после чего DSPL осуществляет реконфигурацию; результаты адаптации сохраняются в базе знаний для обучения.

### **Заключение**

Предложенный подход развивает методы самоадаптации на основе DSPL и моделей характеристик, дополняя их LLM для работы с семантически сложным контекстом. Разработанная архитектура обеспечивает замкнутый цикл самоадаптации, сочетающий интеллектуальный анализ контекста с гарантированно корректной реконфигурацией программы, создавая основу для самоадаптивных систем нового поколения.

### **Список литературы**

1. Бершадский А.М., Бождай А.С., Евсеева Ю.И., Гудков А.А. Разработка комплексного подхода к программной самоадаптации и универсальный метод синтеза адаптивных программных систем // Известия высших учебных заведений. Поволжский регион. Технические науки. — 2021. — № 1 (57). — С. 3–12.
2. Бождай А.С., Евсеева Ю.И., Гудков А.А. Прогнозирование взаимодействий в самоадаптивном программном обеспечении // Известия высших учебных заведений. Поволжский регион. Технические науки. — 2023. — № 1 (65). — С. 75–85.
3. Бождай А.С., Евсеева Ю.И., Гудков А.А. Методы самоадаптации программных систем, основанные на машинном обучении и интеллектуальном анализе данных // Модели, системы, сети в экономике, технике, природе и обществе. — 2021. — № 2 (38). — С. 144–152.
4. da Silva L. M. P. Requirements Engineering and Variability Management in DSPLs Domain Engineering : a systematic literature review / L. M. P. da Silva, C. I. M. Bezerra, R. M. C. Andrade, J. M. S. Monteiro // ICEIS 2016 : proceedings of the 18th International Conference on Enterprise Information Systems, Rome, Italy,

April 25–28, 2016 / ed. S. Hammoudi [et al.]. — Setúbal : SCITEPRESS, 2016. — Vol. 1. — P. 544–551.

5. Benavides D. Concepts / D. Benavides // ISA Research Group, University of Seville [сайт]. — Seville.  
URL: <https://www.isa.us.es/featuremodelanalysis/concepts> (дата обращения: 23.03.2026).
6. Schmid L. Software Architecture Meets LLMs : a systematic literature review / L. Schmid, T. Hey, M. Armbruster, S. Corallo, D. Fuchß, J. Keim, H. Liu, A. Koziolk // KITopen [сайт]. — Karlsruhe, 2025. —  
URL: <https://publikationen.bibliothek.kit.edu/1000181963> (дата обращения: 23.03.2026).
7. Jaring M. On the notion of variability in software product lines / M. Jaring, J. Bosch // Proceedings of the Working IEEE/IFIP Conference on Software Architecture, Amsterdam, Netherlands, August 28–31, 2001. — Piscataway : IEEE, 2001. — P. 45–54.