

УДК 004.89

doi: 10.15622/rcai.2025.103

ПОДХОД К ПОИСКУ КОМПОНЕНТОВ ДЛЯ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ АНАЛИЗА МУЛЬТИМОДАЛЬНЫХ ДАННЫХ¹

С.С. Шевалдов (*sshevaldov@yandex.ru*)

А.А. Филиппов (*al.filippov@ulstu.ru*)

Ульяновский государственный технический университет, Ульяновск

В статье рассматривается решение проблемы поиска компонентов (зависимостей, библиотек) во внешних репозиториях при разработке программных систем с учетом функциональных требований и других особенностей проекта. Важность и сложность данной проблемы объясняется мультимодальной природой данных о компонентах, сложностью интерпретации и оценки параметров компонентов с учетом различных требований и ограничений. Каждый внешний компонент программной системы, имеет ряд параметров, относительно которых разработчик принимает решение о возможности его использования в проекте. Для решения проблемы поиска подходящих компонентов, ускорения процесса выбора наиболее подходящего компонента, а также повышения качества разрабатываемой программной системы предложен подход к поиску компонентов для программных систем на основе анализа мультимодальных данных. Для проверки адекватности предложенного подхода разработан прототип поисковой системы, который обеспечивает сбор, обработку и анализ мультимодальных данных о компонентах с целью получения комплексной оценки для ранжирования результатов.

Ключевые слова: мультимодальные данные, компоненты, программная система, информационный поиск, контекст.

¹ Исследование выполнено в рамках государственного задания № 075-03-2023-143 по проекту «Исследование интеллектуальной предиктивной аналитики на базе интеграции методов конструирования признаков гетерогенных динамических данных для машинного обучения и методов предиктивного мультимодального анализа данных».

Введение

В настоящее время при разработке программных систем для повышения скорости разработки активно применяются принципы компонентно-ориентированного программирования [Nierstrasz, 1992], [Лаврищева, 2018]. Временные затраты сокращаются за счет повторного использования существующих реализаций требуемых модулей и функций. Компоненты обычно оформляются в виде библиотек и помещаются в специализированные хранилища – репозитории [Trew, 2006]. Для различных платформ и/или языков программирования используются отдельные репозитории. В свою очередь компоненты могут использовать возможности других компонентов, в результате чего формируется дерево зависимостей, на основе которого определяется перечень необходимых для подключения к проекту компонентов.

Корректный выбор компонентов является важным шагом процесса проектирования, который существенно влияет на качество разрабатываемой программной системы. Проблема качественного и обоснованного выбора зависимостей обуславливается требованиями к проекту и широким набором характеристик компонентов. При выборе компонентов необходимо учитывать:

- свойства и требования разрабатываемой программной системы;
- квалификацию и опыт разработчиков;
- функциональные возможности, характеристики и качество самих компонентов.

Данные параметры представляют собой множество мультимодальных данных.

Таким образом, возникает задача разработки метода информационного поиска компонентов для программной системы с учетом различных параметров проекта и множества компонентов, представленных набором мультимодальных данных. При этом такая поисковая система должна ранжировать найденные компоненты на основе значения комплексной оценочной величины.

Процесс выбора оптимальных и качественных компонентов с учетом всех ограничений и требований является сложной и ресурсоемкой задачей, т.к. требует анализа данных из различных источников с учетом динамики развития отрасли в целом.

1. Анализ проблемы

В данной работе изучается проблема поиска компонентов для разрабатываемой программной системы, наиболее точно отвечающим динамически изменяющимся требованиям. Мультимодальный характер данных, на основе которых производится выбор компонентов, обусловлен наличием как качественных, так и количественных показателей, с различными методами измерения и представления.

В работе [Перл, 2022] мультимодальные данные определяются как объект или процесс, который описывается различными характеристиками, представленными в соответствующих модальностях. Под модальностью понимается один из нескольких атрибутов, которые описывают конкретную характеристику объекта. Совокупность характеристик одного мультимодального объекта обозначается сущностными представлениями или образами. Каждый мультимодальный объект необязательно имеет сущностное представление во всех модальностях. Признаки могут быть несовместимы друг с другом (например, имя и периодичность). Объект, имеющий несколько описывающих его модальностей, можно назвать мультимодальным.

Важной проблемой является формулировка общего понятия мультимодальных данных. Несмотря на широкую распространённость термина «мультимодальность» в различных науках, часто под ним понимаются разные понятия и явления. Так, в логике принято говорить о модальностях речи и о модальностях, как отражении вероятности высказывания [Nguyen, 2006].

В транспортной логистике говорят о мультимодальности, как о вариативности и комбинировании видов транспорта при доставке грузов [Karimi, 2018].

В биологии и медицине говорят о мультимодальности как о множественном представлении изучаемого или используемого объекта (мультимодальный МРТ, мультимодальная терапия и пр.).

Другим широко распространённым примером мультимодальных данных является наличие разнообразных форматов данных, в первую очередь это изображения, видео и аудио [Перл, 2022].

По своей природе мультимодальные объекты описываются рядом признаков, которые часто не совместимы друг с другом (например, цветом, весом, ценой), и эти характеристики могут рассматриваться как модальности конкретного объекта. В работе [Перл, 2022] определены необходимые виды связей в наборе мультимодальных данных.

Задаваемые извне связи, о которых известно перед началом анализа мультимодальных данных обозначаются как «прямые». Вычисленные связи, отражающие собой меру подобия (силу связывания), обозначаются как «косвенные» связи. В работе [Kalyonova, 2017] подробно рассмотрены связи в мультимодальных данных.

Базовая модель представления мультимодальных данных включает набор аспектов, позволяющих сформировать конечное представление базовой структуры мультимодальных данных. Множество входных данных, разделенных по модальностям, рассматривается в виде

где n – количество уникальных мультимодальных объектов.

Для выполнения анализа полученные наборы данных передаются классификаторам в соответствии с модальностями. Методы классификации зависят от модальностей и не являются четко заданными в рамках всего алгоритма анализа. Классификация проводится с целью определения степени взаимосвязи между представлениями различных мультимодальных объектов.

Результат анализа обычно представляет собой иерархический граф значений в разрезе модальностей. Полученный граф описывает положение конкретной характеристики в общем дереве модальностей.

В рамках поставленной задачи учитываются следующие характеристики компонента:

- Безопасность – сведения об обнаруженных в компоненте уязвимостях (CVE).
- Поддержка проекта – активность разработки, количество незакрытых ошибок, наличие и качество документации.
- Тяжеловесность (зависимости самого компонента) – размер дерева зависимостей компонента с учетом транзитивных связей.
- Совместимость – функциональные и нефункциональные требования, совместимость с платформой и окружением проекта.
- Избыточность – степень использования компонента с учетом его тяжеловесности.
- Популярность – количество использований компонента и степень упоминания компонента в поисковых системах.
- Качество – количество запросов о компоненте с упоминанием различных проблем и ошибок.

Количественными показателями в данном случае являются, к примеру, количество функций компонента, количество незакрытых сообщений об ошибках в репозитории проекта, размер компонента с учетом зависимостей и т. д. Качественными параметрами будут являться: оценка качества поддержки компонента, соответствие требованиям проекта, популярность компонента, и др. Для определения популярности и качества компонента используется анализ динамики количества запросов в поисковых системах с использованием различных шаблонов: «how to», «error», «download» и т. д.

- источниками мультимодальных сведений являются:
- репозитории компонентов для платформы и/или языка программирования (mvn, npm, nuget и др.);
- репозитории компонента, расположенные на IT-хостингах (GitHub, GitLab и др.);
- репозитории сведений об уязвимостях;
- поисковые системы (Yandex, Google и др.);
- большие языковые модели;
- различные сторонние сервисы с полезной статистикой.

Мультимодальные сведения о компонентах позволят реализовать алгоритм информационного поиска компонентов с учетом особенностей и требований конкретного проекта.

Информационный поиск – процесс поиска неструктурированной информации, удовлетворяющей информационные потребности [Закалин, 2018].

Процесс информационного поиска обычно включает в себя последовательность следующих операций, направленных на сбор, обработку и предоставление необходимой информации пользователям с учетом их информационной потребности:

- Уточнение информационной потребности и формулировка запроса.
- Установление совокупности возможных информационных источников.
- Извлечение информации из выявленных информационных источников.
- Ознакомление с полученной информацией и оценка результатов поиска.

В рамках данного исследования информационный поиск компонента рассматривается как комбинация полнотекстового поиска и поиска по метаданным, а информационная потребность пользователя формируется исходя из функциональных и нефункциональных требований проекта и особенностей проекта. Особенности проекта программной системы формируют контекст, в рамках которого учитывается информационная потребность пользователя.

Таким образом, требуется разработка подхода к информационному поиску компонентов для программных систем с учетом мультимодального представления данных и ограничений контекста проекта.

2. Предложенный подход

В данном разделе будет рассмотрен подход к ранжированию компонентов программной системы с учетом степени соответствия информационной потребности на основе анализа мультимодальных данных.

Формально проект программной системы можно представить в виде следующего выражения:

,

где

- компонент программной системы, в котором
- дерево зависимостей i -го компонента;
- множество уязвимостей i -го компонента;
- множество функций i -го компонента;
- информация о динамике популярности i -го компонента;
- информация о процессе разработки i -го компонента;
- метаданные i -го компонента.

На данный момент контекст проекта, который описывает особенности окружения и ограничения проекта, включает только множество функциональных требований. Расширение контекста планируется выполнить в рамках дальнейших исследований.

Тогда, алгоритм поиска компонентов по функциональным требованиям формально можно представить в виде следующей функции:

Функция f осуществляет поиск компонентов C , множество функций которых соответствует функциональным требованиям проекта R .

Функциональные требования проекта R представляют собой текстовое описание, составленное на основе технического задания к разрабатываемой программной системе.

Так как репозитории компонентов содержат большое количество элементов, что затрудняет процесс получение всех данных из такого репозитория, было решено использовать большую языковую модель (LLM) в качестве реализации функции f . В будущем планируется разработка алгоритма для генерации поискового запроса для выбранного репозитория компонентов на основе функциональных требований проекта R .

Для получения множества компонентов C формируется специальный запрос (промпт) для LLM. Для получения промпта используется заранее подготовленный шаблон, в который подставляются указанные функциональные требования R . Шаблон также определяет формат выходных данных для LLM. В итоге ответ LLM содержит в себе перечень имен библиотек, которые в большинстве случаев соответствуют указанным функциональным требованиям. Планируемая интеграция с поисковыми движками существующих репозиториях компонентов позволит сократить количество ошибок работы LLM за счет определения возможных галлюцинаций.

Дальнейшая работа алгоритма поиска компонентов заключается в ранжировании полученного множества компонентов C по заранее установленным правилам.

Формально функцию вычисления комплексной оценки для компонента на основе анализа мультимодальных данных можно представить следующим выражением:

где w_i – веса оценок отдельных параметров компонента;
 f_i – функции для вычисления значений оценки отдельных параметров компонента, представленных мультимодальными данными, на основе нечетких экспертных правил. Вопрос расчета данных параметров выходит за рамки данной статьи и будет рассмотрен в одной из следующих публикаций.

Для проверки адекватности предложенного алгоритма поиска компонентов для программной системы на основе анализа мультимодальных данных был разработан прототип поисковой системы.

3. Архитектура и основные функции поисковой системы

В ходе разработки прототипа поисковой были разработаны следующие подсистемы:

1. Пользовательский интерфейс (UI). Интерфейс для ввода функциональных требований и отображения множества найденных компонентов с указанием оценки. Система позволяет ознакомиться с данными каждого из найденных компонентов, а также получить детальные сведения, на основе которых была получена оценка.

2. Подсистема работы с LLM. Данная подсистема формирует промпт на основе введенных пользователем функциональных требований и осуществляет взаимодействие с LLM.

3. Подсистема загрузки. На основе полученных от LLM имен компонентов данная подсистема производит загрузку данных из внешних источников с последующим их сохранением в хранилище (базе данных). В случае, если данные о компоненте содержатся в хранилище и загрузка данных производилась недавно, производится обращение к подсистеме хранения для их получения.

4. Подсистема хранения. Подсистема хранения позволяет получить ранее загруженные данные о компонентах, а также рассчитать значение комплексной оценки для каждого из них.

5. Хранилище на базе СУБД Postgres.

Для демонстрации примера работы прототипа поисковой системы будем использовать следующий поисковый запрос (функциональные требования): «работа с данными в формате dataframe». В текущей реализации в качестве ограничений контекста указывается язык программирования, в данном случае «python». Контекст учитывается при формировании промпта для LLM.

Например, оценка качества процесса разработки компонента в данной реализации определяется отношением количества закрытых задач (issues) в репозитории проекта компонента к общему числу задач.

В результате работы поисковой системы пользователь получит список компонентов (в данном случае количество ограничено) для указанного языка программирования, отсортированный в порядке убывания вычисленной комплексной оценки.

На рис. 1 представлен результат работы прототипа поисковой системы.

Введите ключевые слова требуемого функционала

dataframe	pip	Найти зависимости			
Имя зависимости	Код последнего релиза	Дата последнего релиза	Ссылка на репозиторий	Количество закрытых issues/общее количество issue, %	Пакетный менеджер
Pandas	2.2.3	2024-09-20	https://github.com/pandas-dev/pandas	86.49	pip
SciPy	1.9.3	2022-10-20	https://github.com/scipy/scipy	84.06	pip
NumPy	2.2.6	2025-05-17	https://github.com/numpy/numpy	82.66	pip

Рис. 1. Пример результата работы прототипа поисковой системы

Как видно из рис. 1, в библиотеке (компоненте) Pandas закрыто 86,49% задач. На основе только лишь этого значения можно сделать вывод о том, что данная библиотека имеет более высокое качество поддержки.

Для расчета оценки популярности компонента была собрана статистика использования компонентов в открытых проектах в рамках IT-хостинга GitHub. В ходе такого анализа производился анализ открытых проектов на языке python с целью выявления их зависимостей (используемых компонентов). В результате была получена статистика о частоте использования различных компонентов. Также была собрана статистика о наиболее популярных комбинациях компонентов в программных системах. Иллюстративный пример полученных статистических данных представлен на рис. 2 и 3.

Целью сбора статистики является определение наиболее популярных библиотек для последующей предварительной загрузки данных о них в хранилище поисковой системы, так как взаимодействие с внешними источниками данных требует значительных временных затрат.

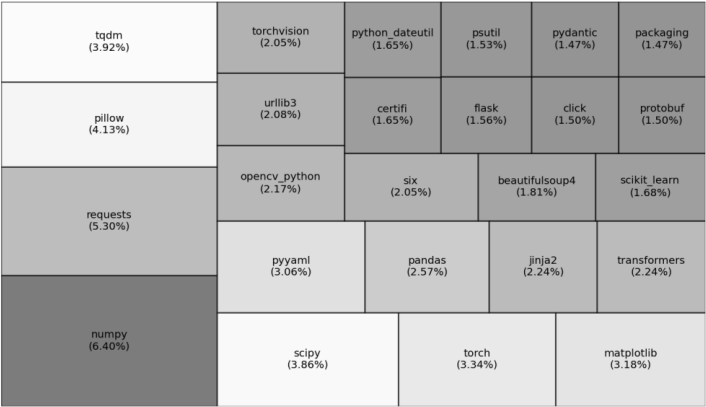


Рис. 2. Диаграмма 25 самых часто используемых компонентов

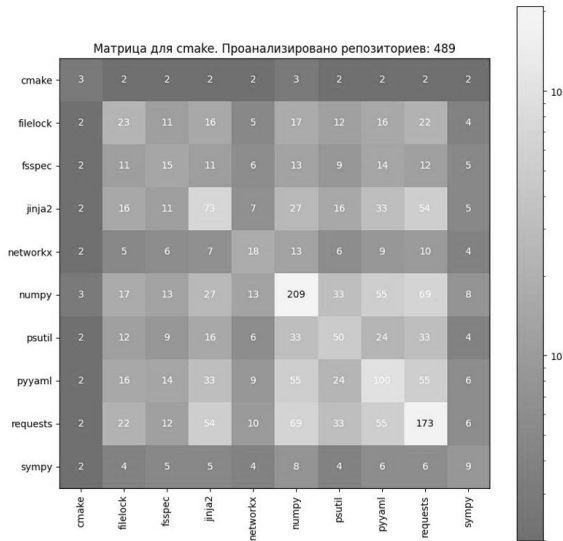


Рис. 3. Матрица комбинаций использования библиотеки «stake»

На рис. 4 представлена ER-диаграмма модели хранилища прототипа поисковой системы.

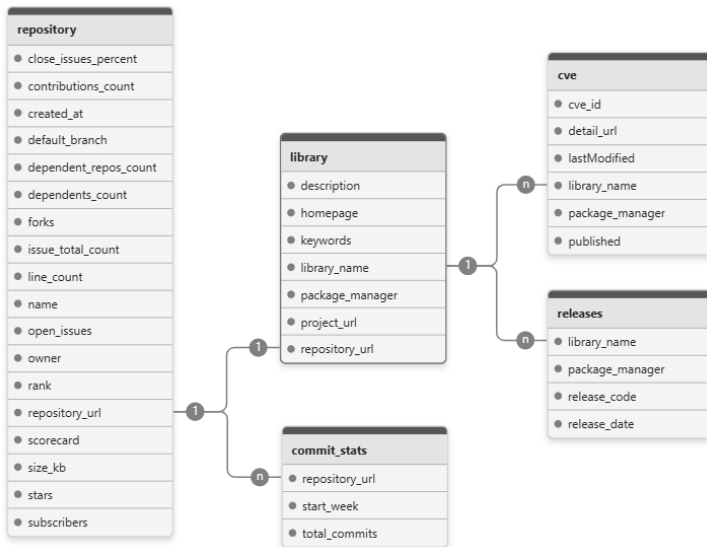


Рис. 4. ER-диаграмма модели данных хранилища поисковой системы

Как видно из рис. 4, сущность Library – основная таблица базы данных хранилища, которая содержит метаданные компонента. Включает в себя имя пакетного менеджера (pip, npm или maven), имя библиотеки, её описание, ссылку на проект и репозиторий проекта на IT-хостинге.

Сущность Repository содержит данные, извлеченные из репозитория проекта на IT-хостинге.

Сущность CVE содержит информацию об уязвимостях компонента.

Сущность Releases содержит информацию о динамике релизов компонента. Учитывается при оценке динамики активности процесса разработки.

В дальнейшем поисковая система будет дорабатываться. Планируется разработка модуля анализа текстов на естественном языке для автоматического извлечения функциональных и нефункциональных требований из текста технического задания.

Заключение

В рамках данной работы предложен подход к сбору данных о компонентах программных систем из различных внешних источников для формирования мультимодального представления. Представлена информационная модель компонента программной системы. Описаны шаги алгоритма поиска компонентов для использования в процессе разработки программной системы на основе функциональных требований, представленных в текстовом виде. Алгоритм поиска основан на автоматическом формировании промпта к LLM с учетом функциональных требований и языка программирования. Также в статье представлен алгоритм расчёта комплексной оценки компонента на основе анализа мультимодальных данных для последующего ранжирования компонентов. Описана архитектура текущей реализации прототипа поисковой системы. Поисковая система реализует все необходимые этапы взаимодействия с пользователем: от получения функциональных требований к компоненту, до вывода пользователю списка найденных компонентов с указанием для каждого из них значения комплексной оценки. Для повышения скорости работы поисковой системы был произведен сбор статистики для определения популярных на данный момент компонентов для проектов на языке python. Список популярных библиотек позволяет загрузить данные о них заранее. В качестве планов дальнейшей работы над проектом можно выделить:

- Разработку подсистемы анализа текста технического задания на разработку программной системы с целью получения перечня функциональных и нефункциональных требований, которые должны быть учтены при поиске компонента.
- Интеграцию с поисковыми механизмами репозитория компонентов для повышения полноты результатов и устранения проблемы галлюцинаций LLM.

Список литературы

- [Nierstrasz, 1992] Nierstrasz O., Gibbs S., Tsichritzis D. Component-oriented software development // Communications of the ACM. – 1992. – Vol. 35, Iss. 9. – P. 160-165.
- [Лаврищева, 2018] Лаврищева Е.М. Программная инженерия. Парадигмы, технологии и case-средства: учебник. 2-е изд.. – М.: Изд-во «Юрайт», 2018. – 280 с.
- [Trew, 2006] Trew T., Soepenbergh G. Identifying technical risks in third-party software for embedded products // Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'05). – IEEE, 2006.
- [Перл, 2022] Перл О.В., Перл И.А. Применение мультимодального подхода для выявления подобия в многомерных наборах данных с примером использования // International Journal of Open Information Technologies. – 2022. – Vol. 10, No. 1. – P. 41-53.
- [Nguyen, 2006] Nguyen L. A. Multimodal logic programming // Theoretical Computer Science. – 2006. – Vol. 360, Iss. 1–3. – P. 247-288.
- [Karimi, 2018] Karimi B., Bashiri M. Designing a Multi-commodity multimodal splittable supply chain network by logistic hubs for intelligent manufacturing // Procedia manufacturing. – 2018. – Vol. 17. – P. 1058-1064.
- [Kalyonova, 2017] Kalyonova O., Perl I. Revealing of entities interconnections in system dynamics modelling process by applying multimodal data analysis paradigm // 2017 21st Conference of Open Innovations Association (FRUCT). – IEEE, 2017. – P. 156-161.
- [Закалин, 2018] Закалин И.Ю. Методы поддержки принятия решений на основе информационных технологий // Вестник магистратуры. – 2018. – №. 1-1(76). – С. 23-26.