

СОДЕРЖАНИЕ

1	ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В КОНТЕКСТЕ ОБЛАСТЕЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	4
1.1	Современная индустрия разработки программного обеспечения	4
1.2	Актуальность мультязыкового статического анализа	5
2	АНАЛИЗ СОСТОЯНИЯ РАБОТ В ОБЛАСТИ АНАЛИЗА МУЛЬТИ-ЯЗЫКОВЫХ ИСХОДНЫХ ТЕКСТОВ ПРОГРАММ	8
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	9
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	10

1 ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В КОНТЕКСТЕ ОБЛАСТЕЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1 Современная индустрия разработки программного обеспечения

Современные программные проекты, в отличие от многих программных проектов прошлого, гораздо чаще состоят из набора разных (порой разительно) технологических решений, предназначенных для решения определенного круга задач. Согласно [1], в одном программном проекте в среднем задействовано 5 языков программирования, один из которых является «основным», а остальные – специализированными языками предметной области (DSL). В заключении статьи авторы признают популярность мультязыковых программных проектов и оценивают важность наличия соответствующих инструментальных средств для работы с такого рода проектами.

Также, нередко использование нескольких языков и в индустрии разработки. Так, авторы [2] провели исследование популярности мультязыковых проектов в результате опроса 139 профессиональных разработчиков из разных сфер. Результаты опроса показали, что опрошиваемые имели дело с 7 различными языками, в среднем. При этом, в работу было вовлечено в среднем 3 пары связанных языков в контексте одного проекта. Более 90% опрошиваемых также сообщали о проблемах согласованности между языками, встречаемых при разработке в такой мультязыковой среде.

Таким образом, в современной разработке программного обеспечения нередко использование нескольких языков вне зависимости от объемов проекта или вовлекаемой предметной области. Ситуация становится сложнее со временем, так как создание новых технологий разработки часто влечет за собой формирование определенной нотации или языка для управления или конфигурации. Например, это может касаться таких повсеместных технологий как СУБД, система сборки, сервер приложений или скрипты развертывания.

Для наглядности, можно привести следующие языки, нередко фигурирующие в составе современных программных проектов:

- язык разметки HTML в составе проекта, использующего ASP фреймворк,
- язык скриптов командной строки в составе проекта, использующего язык C,
- язык запросов SQL в составе проекта, использующего Python и фреймворк Flask,
- язык препроцессора в составе файла исходного кода, реализованного на C++.

Заключительный пункт списка примеров приведен для того, чтобы показать характер связи различных технологий — разным языкам необязательно даже находится в отдельных файлах или модулях, нередко случаи полноценного переплетения различных синтаксисов и семантик.

1.2 Актуальность мультязыкового статического анализа

Итак, мультязыковые программные проекты нередки. Следовательно, имеет смысл использования различных техник работы с исходным кодом таких проектов, поддерживающих процесс разработки. Одной из таких техник, обеспечивающей разные сценарии использования, является статический анализ исходного кода.

Стоит уточнить что имеется под определением «статический анализ кода». Статический анализ это прежде всего набор различных техник по извлечению информации о программе без явного её запуска [3]. Таким образом, статический анализ может быть полезен в сценариях, которые не предполагают явного запуска программы – в сущности во всех сценариях процесса разработки ПО исключая этапы тестирования и запуска.

Возможные сценарии использования статического анализа кода включают (но не ограничиваются):

- оптимизацию программ,
- выявление потенциальных уязвимостей,

- доказательство сохранения определенных инвариантов,
- сбор определенной статистики,
- выявление «пахнущих» фрагментов кода,
- помощь разработчику во время кодирования,
- автоматический рефакторинг кода.

Так как сценарии использования статического анализа настолько разнообразны, в рамках данной работы решено было сосредоточиться на аспектах разработки, которые помогают в процессе кодирования и поддержки проекта. В качестве прикладной реализации таких аспектов выступают различные инструментальные средства. К таким средствам, к примеру, относятся:

- интегрированные средства разработки (IDE),
- линтеры (собирает название инструментов, первым из которых был «Lint» [4]),
- инструменты автоматического рефакторинга,
- инструменты сбора статистики,
- различные кодогенераторы и фреймворки [5][6].

Необходимость в таких средствах присутствует и она достаточно высока. Так, согласно исследованию [7], использование средств поддержки разработчика (в данной работе это механизмы анализа и навигации по межъязыковым связям) позволяет улучшить как скорость разработки ПО, так и уменьшить количество совершаемых ошибок. Стоит заметить, что несмотря на количество времени, прошедшее с момента проведения исследования, принципы разработки ПО в данной предметной области (веб-разработка) не изменились и большинство программных проектов веб-приложений состоят как минимум из двух языков. Обычно это разделение проводится по принципу фронтенд и бекенд.

Также, в последние годы тема исследования межъязыковых зависимостей для обеспечения инструментального анализа начала освещаться более подробно. Так, в статье [9] рассматриваются 76 исследований ориентированных на выявление межъязыковых связей в различных предметных областях. Авторы признают важность систематического обзора данной темы как в контексте

разработки и поддержки бизнес решений, так и в отношении академических исследований. Соответственно, существует практическая проблема сопровождения мультязыковых систем, которая присутствует как в промышленном ПО, так и в академических разработках.

2 АНАЛИЗ СОСТОЯНИЯ РАБОТ В ОБЛАСТИ АНАЛИЗА МУЛЬТИЯЗЫКОВЫХ ИСХОДНЫХ ТЕКСТОВ ПРОГРАММ

<https://arxiv.org/pdf/1906.00815.pdf>

<https://ieeexplore.ieee.org/abstract/document/7953501>

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ПО — программное обеспечение

СУБД — система управления базами данных

LSP — language server protocol

IDE — integrated development invironment

AST — abstract syntax tree

DSL — domain specific language

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. P. Mayer and A. Bauer, «An empirical analysis of the utilization of multiple programming languages in open source projects», in Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, Nanjing, China, 2015.
2. Mayer, P., Kirsch, M. & Le, M.A. On multi-language software development, cross-language links and accompanying tools: a survey of professional software developers. J Softw Eng Res Dev 5, 1 (2017). <https://doi.org/10.1186/s40411-017-0035-z>
3. Static Program Analysis Anders Møller and Michael I. Schwartzbach Department of Computer Science, Aarhus University. May 2023. <https://cs.au.dk/~amoeller/spa/>
4. Johnson, Stephen C. (25 October 1978). "Lint, a C Program Checker". Comp. Sci. Tech. Rep. Bell Labs: 78–1273. CiteSeerX 10.1.1.56.1841.
5. <https://doc.qt.io/qt-6/metaobjects.html>
6. <https://react.dev/>
7. Pfeiffer, R.H., Wąsowski, A. (2012). Cross-Language Support Mechanisms Significantly Aid Software Development. In: France, R.B., Kazmeier, J., Breu, R., Atkinson, C. (eds) Model Driven Engineering Languages and Systems. MODELS 2012. Lecture Notes in Computer Science, vol 7590. Springer, Berlin, Heidelberg. pp 168-184 https://doi.org/10.1007/978-3-642-33666-9_12
8. J. M. Fernandes, G. H. Travassos, V. Lenarduzzi, and X. Li, Quality of Information and Communications Technology: 16th International Conference, QUATIC 2023, Aveiro, Portugal, September 11–13, 2023, Proceedings. Springer Nature Switzerland, 2023.
9. S. Latif, Z. Mushtaq, G. Rasool, F. Rustam, N. Aslam, and I. Ashraf, 'Pragmatic evidence of cross-language link detection: A systematic literature review', Journal of Systems and Software, vol. 206, p. 111825, 2023.