

НОВЕЙШИЙ СПОСОБ ОБМЕНА МЕТРИКАМИ

И.А. Хомяков

¹ Университет Иннополис, г.Казань

Сбор метрик программного обеспечения является фундаментальной деятельностью, которая необходима для проведения практически любого эмпирического исследования в области программной инженерии. Однако, даже при наличии широкого спектра инструментов, сбор таких фундаментальных данных по-прежнему занимает много времени. Более того, каждый исследователь собирает практически одни и те же данные (например, метрики СК, цикломатическая сложность МакКейба и т.д.) из практически одних и тех же проектов (например, из известных проектов с открытым исходным кодом). Объем такой дублирующей работы, выполняемой в сообществе, уменьшает усилия, которые исследователи могут потратить на наиболее ценную часть своих исследований, такую как разработка новых теорий и моделей и их эмпирическая оценка. В данной работе предлагается новый подход для сбора и обмена данными метрик программного обеспечения, позволяющий сотрудничать исследователям и сократить количество напрасных усилий в сообществе разработчиков программного обеспечения. Мы стремимся достичь этой цели, предлагая Формат обмена программными метриками (SMEF) и REST API для сбора, хранения и обмена данными метрик программного обеспечения.

Ключевые слова: метрики программного обеспечения, формат обмена, обмен метриками.

ВВЕДЕНИЕ

Постоянный сбор и анализ метрик исходного кода являются очень важными для отслеживания и управления процессом разработки программного обеспечения [1] [2]. Для удовлетворения данных потребностей было разработано несколько метрик программного обеспечения, и их число постоянно растет [3] [4]. В частности, из-за специфических наборов метрик, разработанных для удовлетворения потребностей конкретных процессов (например, agile-методов) и/или категорий проектов (например, open source). Более того, согласно данным Нунес-Варела и др [3], существует по крайней мере 14 широко используемых инструментов метрик программного обеспечения. Из этого числа естественно, исключены исследовательские прототипы и инструменты, которые собирают узкий набор метрик, которые используются ограниченным кругом людей. Однако найденные инструменты имеют ряд ограничений, в том числе: - Для поддержки разных языков и/или метрик необходимы разные инструменты. - Они не могут быть легко расширены для поддержки новых метрик и/или языков. - При сравнении результатов работы различных инструментов результаты оказываются противоречивыми. Поэтому очевидно, что интеграция и обмен информацией между различными инструментами необходимы как исследователям, так и специалистам-практикам [5] [6]. Существует большое количество работ, посвященных проблеме формализации программного обеспечения метрик, начиная с 1991 года [7] и до настоящего времени [8]. Такие работы можно разделить на две основные категории: Описание внутренней структуры метрик: в таких исследованиях авторы фокусируются на определении метамodelей для описания метрик и способов их расчета. Описание внешней структуры метрик: в таких исследованиях

авторы фокусируются на обмене и хранении метрических данных. **Описание внутренней структуры метрик** Первая работа, связанная с формализацией метрик [9], дает формальное определение для нескольких метрик, включая одну для измерения размера программного обеспечения. Позже в [10] был предложен подход, основанный на грамматике, где атрибуты измерений привязываются к определениям языка. В нескольких последующих работах эта проблема рассматривалась через определения различных метамodelей и механизмов запросов для извлечения данных метрик. Например, в [11] были созданы определения метрик программного обеспечения, основанные на языке объектных ограничений (OCL), метамodelи GOODLY и UML Маквиллан и Пауер [12] [13] расширили более раннюю работу Варони и др. [11] для создания языка на основе OCL для определения метрик. Эль-Вакиль и др. [14] использовали метамodelь ODEM [15] и XQuery для представления формул оценки метрик. Они использовали XML DOM дерево с XQuery для обработки дерева для оценки значений для метрик. Другие подходы можно найти в [16], где мета-модель программных метрик (SMM) используется для генерации метрик. Кроме того, Линке и Лёве [17] [18] показали структуру для определения метрик программного обеспечения, основанную на абстрактной метамodelи. Одной из последних работ на эту тему является [19]. В ней описывается формат, который позволяет определять существующие метрики и даже представлять любые новые. Однако большинство из перечисленных здесь работ страдают от ограничений связанные с конкретной принятой моделью и часто ограниченные определенным языком. Более того, некоторые из них способны охватить только проектные отношения. Наконец, в них не рассматривается возможность повторного использования определения метрик, что позволило бы

избежать переопределения одних и тех же метрик снова и снова.

Описание внешней структуры метрики

Наиболее перспективное решение проблем, связанных с обменом метриками, описано в [20]. Авторы представляют протокол обмена, основанный на XML и технологии веб-сервисов SOAP. Однако предложенное решение является довольно старым и имеет ряд ограничений: Документ не содержит полного и подробного описания предлагаемого формата обмена XML. Протокол описан только на очень высоком уровне без деталей, необходимых для фактической реализации. Протокол основан на SOAP, что делает его довольно тяжеловесным по сравнению с существующими подходами. Не существует никаких практических реализаций. Не существует публикаций, непосредственно расширяющих данную работу. Работа Харрисона [21] фокусируется на проблеме хранения и извлечения метрических данных в едином хранилище, но не учитывает не учитывает проблемы, связанные с процессом обмена такими данными

Проблемы, вдохновившие авторов статей, приведенных выше, являются еще более актуальными в настоящее время в связи с большим количеством исследований в области эмпирической программной инженерии, растущих потребностей в данных для проведения таких исследований, а также доступности все большего количества несовместимых инструментов. По этим причинам мы считаем, что существует необходимость в новой технологии для сбора, хранения и обмена данными о метриках программного обеспечения. Для этого мы определили очень простую архитектуру, в которой каждый инструмент извлечения метрик сопряжен с конвертером формата, который способен перевести специфический формат файла инструмента в SMEF. Затем, преобразованный файл может быть отправлен на обмен или сделан общедоступным, сохраняя его в общем хранилище.

| Метод | Путь | Параметры | Функционал |
|-------|----------------------|-------------------------------|--|
| GET | /rest/api/v1/regions | projectUrl, revision | Получение региона |
| GET | /rest/api/v1/metrics | ProjectID, regionID | Получение блока метрик |
| GET | /Regions/ | Timestamp RegionID code | Возвращает все метрики с указанными параметрами |
| POST | /SMEF | | Отправляет данные в базу данных в виде файла (формат SMEF xml) |

По указанным причинам исследователи вынуждены выполнять извлечение данных из одних и тех же хранилищ снова и снова. Такие повторные извлечения имеют смысл только в том случае, если

извлекаемые данные являются новыми (либо новые виды метрик или уже существующие метрики из различных проектов или версий). Во всех остальных случаях они являются пустой тратой ценных ресурсов, которые могут быть направлены на улучшение технологического уровня. Когда исследователи используют инструменты сбора метрик, возникает ряд проблем, с которыми им приходится сталкиваться. В частности, несовместимость извлеченных данных с точки зрения семантики и синтаксиса, что делает обмен информацией крайне сложным. Проблемы, связанные с семантикой, связаны с разницей в определениях некоторых метрик. Типичным примером является количество строк кода (LOC), которое может быть подсчитано множеством различных способов (например, включая или исключая пустые строки), при этом все они очень популярны и используются в различных инструментах. Проблемы, связанные с синтаксисом, связаны с различными форматами файлов, которые используют инструменты извлечения. Все они используют различные способы (например, обычный текст, HTML, XML, JSON и т.д.). Подобные проблемы затрагивают и практикующих специалистов, поскольку метрики ценны во многих промышленных контекстах по нескольким причинам, в том числе: Оценка проектов с открытым исходным кодом: проекты с открытым исходным кодом широко разрабатываются и используются практически в любом промышленном контексте, и легкий доступ к данным метрик может помочь практикам быстро и легко проверять соответствующие свойства проекта (например, качество, сопровождаемость, эволюция и т.д.). Непрерывная внутренняя оценка: поскольку имеющиеся для извлечения метрик не способны обмениваться данными, анализ внутренних проектов может быть затруднен, особенно если специалисты заинтересованы в сборе различных наборов метрик что требует использования разнообразных инструментов. Это барьер для практического применения самых передовых и новейших методов, разработанных исследователями.

Данная работа направлена на решение проблемы сбора и обмена метриками, предлагая интегрированную систему, которая включает в себя следующие компоненты. –1. Software Metrics Exchange Format (SMEF): протокол обмена данными на основе XML, который позволяет кодировать данные метрик недвусмысленным образом 2. Metricspedia: каталог метрик программного обеспечения с подробным описанием их извлечения и ссылками на соответствующие научные публикации.– 3. SoftwareMetricsHub: репозиторий метрик программного обеспечения с REST API для хранения, обмена и получения данных метрик.

ОСНОВНАЯ ЧАСТЬ

Основываясь на работе Ауэра [20] и анализе ряда ограничений существующих подходов, описанных в предыдущих разделах. Мы предлагаем формат данных для обеспечения обмена метриками программного обеспечения, который должен выступать в качестве универсального языка для таких данных. Основная идея предлагаемой системы заключается в том, что поскольку каждый инструмент извлечения метрик собирает данные, используя свой собственный формат данных и собственную семантику метрик, необходимо: 1) Определить общий формат представления. 2) Определить соответствие между метриками, специфичными для конкретного инструмента, и общим каталогом метрик, которые определены однозначно и находятся в открытом доступе (например, различные способы подсчета строк кода (LOC) фактически сопоставлены с различными метриками). Таким образом, можно будет иметь общий формат файла и идентифицировать данные, относящиеся к одной и той же метрике, даже если они извлечены с использованием различных инструментами и имеют разные названия. В качестве первой черновой версии формата мы решили использовать XML схему для его описания, поскольку она предоставляет очень мощный инструмент для определения ограничивающих условий, а XML-документы легко поддаются валидации. Мы также рассмотрели возможность принятия более современного формата такого как JSON, однако схема JSON, которая может быть использована для валидации документов для обеспечения их правильности, находится все еще находится в стадии разработки, и она гораздо менее мощная по сравнению с XML Schema. Общая структура описана на рисунке 1. SMEF требует не только, чтобы данные были закодированы с помощью XML файл, соответствующий описанной схеме, но и всеобъемлющий каталог недвусмысленных определений метрик, который в настоящее время в стадии разработки и будет называться SoftwareMetricspedia. Формат содержит несколько ключевых элементов, описанных в следующих подразделах. SMEF это корневого элемент документа, который содержит основную информацию об анализируемом проекте. Основная идея заключается в том, чтобы создать специальный контейнер, определяющий конкретную версию проекта, который анализируется, а также инструмент который используется для извлечения информации о проекте. Последняя информация также может быть использована для проверки согласованности извлеченных метрик в различных инструментах, что является проблемой, которая часто возникает в популярных инструментах, как описано в разделе 2. Внутри этого корневого элемента находится список

Регионов. Регион Мы разделили программный проект на регионы кода. Регион кода может быть любым видом сегмента кода (например, весь проект, файл, модуль, класс, метод и т.д.) в соответствии с конкретным языком программирования. Регионы организованы в иерархическую структуру, начиная с корня, который определяется как весь проект. Затем перечисляются все остальные регионы со ссылкой на родительский регион. Таким образом, XML-документ имеет довольно плоскую структуру, и можно легко восстановить структуру проекта, просто перемещаясь по всем регионам и следуя их ссылкам. С каждым регионом кода можно связать Метрики. Метрика. Чтобы избежать путаницы и неправильного толкования данных, каждая метрика, поддерживаемая SMEF, включена в каталог (Metricspedia), который подробно описывает метрику и способ расчета. Этот каталог будет использоваться для разграничения различных версий аналогичных метрик (например, различных версий предоставления строк кода), которые имеют различные правила расчета. Уникальный идентификатор, хранящийся в каталоге, используется для идентификации конкретной метрики и связывания извлечённого числового значения. Входной параметр Некоторые регионы не могут быть однозначно-но идентифицированы по их именам (например, методы), но они требуют идентификации их сигнатуры через их входные параметры (это обычно требуется для перегруженных методов). По этой причине для таких регионов необходимо описать их входные параметры, которые будут использоваться для их уникальной идентификации. Очень простой пример реализации описанной схемы приведен в Листинге 2

SoftwareMetricspedia. Это вики-сайт, работающий на движке MediaWiki который включает в себя описание большинства метрик кода, доступных в литературе. В настоящее время сайт находится в стадии разработки, и он будет являться отправной точкой для SMEF, где будут зарегистрированы все идентификаторы метрик, и она может быть использована для однозначного определения метрик. Более того, как и в любой другой вики, сообщество сможет вносить свой вклад в определение новых метрик и/или улучшать уже существующие.

SoftwareMetricsHub. Это публичное хранилище метрик, наполняемое с помощью файлов SMEF, которыми делится сообщество. Он реализован как простая база данных с лёгким в использовании программным интерфейсом, который может быть использован с помощью различных технологий и платформ. Для достижения этой цели был разработан простой REST API. API позволяет вставлять и извлекать структуру проектов (в терминах регионов и их зависимостей) и метрических данных, связанных с каждым регионом

ЗАКЛЮЧЕНИЕ

В данной статье был представлен краткий обзор SMEF, формата обмена метриками программного обеспечения и его вспомогательной инфраструктуры для поддержки продвижения обмена данными метрик программного обеспечения. Мы считаем, что развитие такой инфраструктуры будет полезно для исследователей и практиков, чтобы сосредоточиться на анализе данных, а не на их сборе. Мы осознаем, что формат и инфраструктура находятся лишь на стадии чернового варианта и требуют дальнейшего развития при поддержке всего сообщества. В частности, большая часть усилий будет направлена на создание каталога метрик и REST API, включая возможность проведения базового анализа

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Irina Diana Coman and Alberto Sillitti. An empirical exploratory study on inferring developers' activities from low-level data. In 19th International Conference on Software Engineering and Knowledge Engineering (SEKE 2007), July 2007.
2. I. D. Coman, P.N. Robillard, A. Sillitti, and G. Succi. Cooperation, collaboration and pair-programming: Field studies on backup behavior. *Journal of Systems and Software*, 91(5), May 2014.
3. Alberto Salvador Núñez Varela, Héctor Gerardo Pérez-González, Francisco E. Martínez-Pérez, and Carlos Soubervielle-Montalvo. Source code metrics: A systematic mapping study. *J. Syst. Softw.*, 128:164–197, 2017.
4. Anton Bykov, Vladimir Ivanov, Alan Rogers, Alexandr Shunevich, Alberto Sillitti, Giancarlo Succi, Alexander Tormasov, Jooyong Yi, Albert Zabirov, and Denis Zaplatnikov. A new architecture and implementation strategy for non-invasive software measurement systems. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing. ACM, apr 2018.
5. Valentina Lenarduzzi, Alberto Sillitti, and Davide Taibi. A survey on code analysis tools for software maintenance prediction. In SEDA, 2018.
6. Valentina Lenarduzzi, Alberto Sillitti, and Davide Taibi. Analyzing forty years of software maintenance models. 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), pages 146–148, 2017.

7. Lem O. Ejiogu. Tm: a systematic methodology of software metrics. *ACM SIGPLAN Notices*, 26:124–132, 1991.
8. M. Alshayeb, Yasser Shaaban, and J. AlGhamdi. Spmdl: Software product metrics definition language. *ACM J. Data Inf. Qual.*, 9:20:1–20:30, 2018.
9. Lem O. Ejiogu. Tm. *ACM SIGPLAN Notices*, 26(1):124–132, янв 1991.
10. B.I. Cogan and R.B. Hunter. Language-based approaches to software measurement. In Proceedings of the 3rd International Software Metrics Symposium, pages 3–9, 1996.
12. Jacqueline A. McQuillan and J. F. Power. A definition of the chidamber and kemerer metrics suite for uml. 2006.
13. Jacqueline A. McQuillan and James F. Power. Experiences of using the dagstuhl middle metamodel for defining software metrics. In Proceedings of the 4th International Symposium on Principles and Practice of Programming in Java, PPPJ '06, page 194–198, New York, NY, USA, 2006. Association for Computing Machinery.
14. M. El-Wakil, Ali El-Bastawisi, and M. Riad. A novel approach to formalize object-oriented design metrics. 2005.
15. R. Reißing. Towards a model for object-oriented design measurement. 1996.
16. C. Hein, Marcus Engelhardt, Tom Ritter, and Michael Wagner. Generation of formal model metrics for mof based domain specific languages. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, 24, 2009.
17. Rüdiger Lincke and Welf L'öwe. Foundations for defining software metrics. 2006.
18. D. Strein, Rüdiger Lincke, Jonas Lundberg, and Welf L'öwe. An extensible meta-model for program analysis. 2006 22nd IEEE International Conference on Software Maintenance, pages 380–390, 2006.
19. Mohammad Alshayeb, Yasser Shaaban, and Jarallah Al-Ghamdi. Spmdl. *Journal of Data and Information Quality*, 9(4):1–30, май 2018.
20. M. Auer. Measuring the whole software process : A simple metric data exchange format and protocol. 2002.
21. W. Harrison. A flexible method for maintaining software metrics data: a universal metrics repository. *J. Syst. Softw.*, 72:225–234, 2004.

Хомяков Илья Антонович – аспирант, Университет Иннополис, тел. +79991568474, e-mail: I.khomyakov@innopolis.ru.

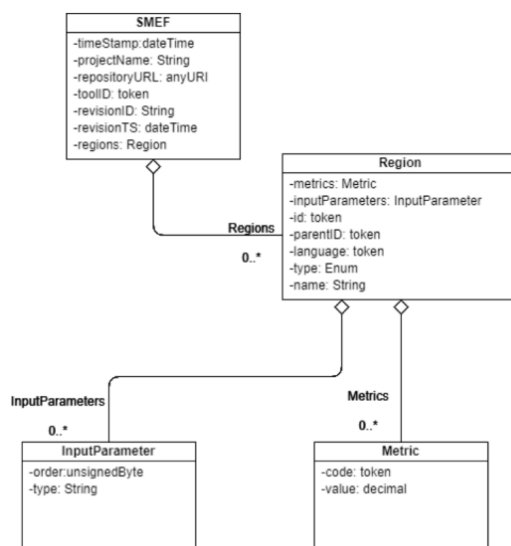


Рис. 1. Структура формата обмена метриками

NOVEL APPROACH FOR SOFTWARE METRICS SHARING

I.A. Khomyakov¹

¹Innopolis University Innopolis

In almost every empirical software engineering study, software metrics collection is a fundamental activity. Although many tools exist to collect this data, it still takes a considerable amount of time. In addition, almost all researchers collect essentially the same data (e.g., CK metrics, McCabe Cyclomatic Complexity, etc.) from essentially the same sources (e.g., well-known open-source projects). Having so much duplication of work done within a community reduces the amount of time that researchers can spend developing new ideas and evaluating them empirically, which is the most valuable part of their research. In this paper, we propose a novel approach for getting and sharing software metrics data that will allow them to collaborate and reduce the amount of wasted effort. SMEF, a file format for exchanging software metrics information, and a REST API, targeted at this objective, are proposed in this paper.

Index terms: new exchange format, software metrics

REFERENCES

1. Irina Diana Coman and Alberto Sillitti. An empirical exploratory study on inferring developers' activities from low-level data. In 19th International Conference on Software Engineering and Knowledge Engineering (SEKE 2007), July 2007.
2. I. D. Coman, P.N. Robillard, A. Sillitti, and G. Succi. Cooperation, collaboration and pair-programming: Field studies on backup behavior. *Journal of Systems and Software*, 91(5), May 2014.
3. Alberto Salvador Núñez Varela, Hector Gerardo Pérez-González, Francisco E. Martínez-Pérez, and Carlos Soubervielle-Montalvo. Source code metrics: A systematic mapping study. *J. Syst. Softw.*, 128:164–197, 2017.
4. Anton Bykov, Vladimir Ivanov, Alan Rogers, Alexandr Shunevich, Alberto Sillitti, Giancarlo Succi, Alexander Tormasov, Jooyong Yi, Albert Zabirov, and Denis Zaplatnikov. A new architecture and implementation strategy for non-invasive software measurement systems. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM, apr 2018.
5. Valentina Lenarduzzi, Alberto Sillitti, and Davide Taibi. A survey on code analysis tools for software maintenance prediction. In *SEDA*, 2018.
6. Valentina Lenarduzzi, Alberto Sillitti, and Davide Taibi. Analyzing forty years of software maintenance models. *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 146–148, 2017.
7. Lem O. Ejiogu. Tm: a systematic methodology of software metrics. *ACM SIGPLAN Notices*, 26:124–132, 1991.
8. M. Alshayeb, Yasser Shaaban, and J. AlGhamdi. Spmdl: Software product metrics definition language. *ACM J. Data Inf. Qual.*, 9:20:1–20:30, 2018.
9. Lem O. Ejiogu. Tm. *ACM SIGPLAN Notices*, 26(1):124–132, янв 1991.
10. B.I. Cogan and R.B. Hunter. Language-based approaches to software measurement. In *Proceedings of the 3rd International Software Metrics Symposium*, pages 3–9, 1996.
11. Jacqueline A. McQuillan and J. F. Power. A definition of the chidamber and kemerer metrics suite for uml. 2006.
12. Jacqueline A. McQuillan and James F. Power. Experiences of using the dagstuhl middle metamodel for defining software metrics. In *Proceedings of the 4th International Symposium on Principles and Practice of Programming in Java, PPPJ '06*, page 194–198, New York, NY, USA, 2006. Association for Computing Machinery.
13. M. El-Wakil, Ali El-Bastawisi, and M. Riad. A novel approach to formalize object-oriented design metrics. 2005.
14. R. Reißing. Towards a model for object-oriented design measurement. 1996.
15. C. Hein, Marcus Engelhardt, Tom Ritter, and Michael Wagner. Generation of formal model metrics for mof based domain specific languages. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, 24, 2009.
16. Rüdiger Lincke and Welf Löwe. Foundations for defining software metrics. 2006.
17. D. Strein, Rüdiger Lincke, Jonas Lundberg, and Welf Löwe. An extensible meta-model for program analysis. *2006 22nd IEEE International Conference on Software Maintenance*, pages 380–390, 2006.
18. Mohammad Alshayeb, Yasser Shaaban, and Jarallah Al-Ghamdi. Spmdl. *Journal of Data and Information Quality*, 9(4):1–30, май 2018.
19. M. Auer. Measuring the whole software process: A simple metric data exchange format and protocol. 2002.
20. W. Harrison. A flexible method for maintaining software metrics data: a universal metrics repository. *J. Syst. Softw.*, 72:225–234, 2004.

Khomyakov Ilya Antonovich – Phd student, Innopolis University, +7 (843) 203-92-53, e-mail: l.khomyakov@innopolis.ru.