

А. С. Югов

## О применении анализа логов событий для обнаружения антипаттернов в сервис-ориентированных системах

**АННОТАЦИЯ.** Сервис-ориентированные системы, как и любые другие программные системы, развиваются с течением времени. Независимо от того, какими были предпосылки изменений: новые требования, изменение среды функционирования, и т.п.

Постоянные изменения могут привести к появлению в системе «плохих» решений – антипаттернов, что, в свою очередь, снижает качество программной системы и требует большего внимания разработчиков при поддержке и дальнейшем развитии продукта. В данной статье рассмотрены примеры антипаттернов, а также способы их автоматизированного нахождения с помощью построения модели реального поведения системы на основе методов обнаружения процессов (Process Mining).

*Ключевые слова и фразы:* сервис; обнаружение антипаттернов; Process Mining

### Введение

Использование сервисов при построении программных систем является на сегодняшний день широко распространенной практикой. А постоянные изменения создают риски внесения антипаттернов в систему.

Исследования [8] показывают, что использование при создании системы паттернов проектирования ненамного снижает трудозатраты на поддержание системы. Однако, наличие антипаттернов значительно увеличивает трудозатраты на поддержку. Таким образом, данная работа нацелена именно на обнаружение антипаттернов в сервис-ориентированных системах.

Встает вопрос, как отследить все вносимые изменения, а, в идеале, отслеживать изменение состояния системы в процессе ее функционирования. Способом, позволяющим решить такую задачу, является анализ логов событий. Лог содержит всю информацию о действиях системы, к тому же, постоянно пополняется новыми данными, т.е. всегда содержит актуальную информацию. Поэтому мы будем строить анализ, опираясь на логи событий и методы, применяемые в области Process Mining.

Далее в работе представлены введение в сферы обнаружения антипаттернов и извлечения знаний из логов событий. После чего описан предлагаемый подход для автоматизированного обнаружения антипаттернов.

## 1. Развитие обнаружения антипаттернов

Понятие качества необходимо для создания продуктов, которые легко поддерживать и развивать. Наличие паттернов и антипаттернов признано одним из наиболее эффективных способов измерения качества различных систем [5].

Первые попытки обнаружения антипаттернов были предприняты в отношении объектно-ориентированных (ОО) систем. Стоит отметить, что данное направление довольно хорошо проработано, и существуют готовые методики для поиска как паттернов, так и антипаттернов [2], [3].

Однако, далеко не все подходы, используемые для ОО систем, могут быть применены к сервис-ориентированным (СО) системам. Одним из таких примеров является DÉCOR [4], предложенный изначально для ОО сред, позднее был адаптирован для СО [9]. Подход основан на расчете заранее определенных метрик, набор которых со специфически подобранными значениями идентифицируют антипаттерн. На основе подхода «DÉCOR» был разработан инструмент SODA (Service Oriented Framework for Antipatterns) [6].

Однако, у SODA есть свои ограничения. Для работы SODA необходим доступ к исходному коду. Как следствие, с помощью SODA

можно анализировать только проекты с открытым исходным кодом. Доступ к исходному коду необходим для инструментирования кода специальными конструкциями, которые собирают метрики, необходимые для обнаружения антипаттернов. Соответственно, для подготовки проведения анализа какой-либо системы требуются дополнительные трудозатраты.

Другой инструмент автоматического обнаружения антипаттернов – SPARSE [10]. В SPARSE, антипаттерны представляются в виде онтологии в формате OWL (Web Ontology Language), а правила по обнаружению описываются с помощью SWRL (Semantic Web Rule Language).

Есть несколько работ, которые специализируются на обнаружении достаточно специфических антипаттернов, например, связанных с показателями производительности или потреблении ресурсов. Например, Вонг [13] описывает применение генетических алгоритмов для обнаружения аномального поведения системы с точки зрения потребления ресурсов (память, время процессора, количество потоков). Подход основан на расчете специализированных для потребления ресурсов метрик.

## **2. Извлечение знаний из логов процессов и Process Mining**

Большое количество исследований посвящено извлечению знаний из логов процессов. В рамках данной работы интересны исследования, объектами изучения которых являются сервисы.

Первая работа [11] сосредоточена на выявлении паттернов композиции сервисов, т.е. множеств сервисов, которые постоянно вызываются вместе, а также похожи по структуре и функционалу.

Некоторые проекты сосредоточены на обнаружении паттернов взаимодействия с помощью анализа логов событий. В работе Ка-Уе Ng и др. [7] был предложен подход для обнаружения порождающих и поведенческих паттернов проектирования в ОО системах. Авторы предлагают инструментировать байт-код и восстановить диаграммы сценариев. После этого авторы распознают паттерны на основе взаимодействий в процессе выполнения программы.

Другая работа в том же направлении представляет идентификацию паттернов проектирования по логам событий, применяя воспроизведение сценариев [1]. Данный подход предполагает выполнение специализированных сценариев, что, в некотором смысле, противоречит общему подходу к анализу процессов. Подход предназначен так же для ОО систем.

Несмотря на большую схожесть ОО и СО подходов, методы поиска (анти)паттернов, предназначенные для первого, не могут быть легко адаптированы ко второму, т.к. СО подход предполагает разбиение на отдельные функциональные модули, а также является более динамичным по своей природе. Эти различия приводят к тому, что во внимание должны быть приняты дополнительные аспекты, которые не были важны при работе с ОО системами.

Когда мы говорим об анализе логов событий, то нельзя не упомянуть Process Mining – относительно новую дисциплину, предоставляющую обширный набор инструментов для обеспечения анализа процессов на основе их реального поведения (строится по логам) и дальнейшего совершенствования процессов [12]. Методы Process Mining объединяют в себе модели процессов и данные о событиях, что делает возможным выявление отклонений, предсказание задержек, поддержку принятия решений и предоставление рекомендаций по улучшению процессов.

### **3. Обнаружение антипаттернов в сервис-ориентированных системах**

Практически во всех упомянутых выше работах обнаружение антипаттернов производилось с помощью расчета метрик. В данной работе для обнаружения антипаттернов также будет использован подход на основе метрик.

Далее представлены метрики, используемые в данной работе:

- (1) количество входящих вызовов сервиса (кто-то вызывает данный сервис);

- (2) количество исходящих вызовов сервиса (данный сервис вызывает кого-то);
- (3) время ответа;
- (4) количество взаимных вызовов;
- (5) связность с другими сервисами; и др.

Для вычисления значений каждой из метрик строится специфическая для нее модель. Конкретные значения, полученные в процессе вычисления показывают, входит ли сервис в антипаттерн или нет.

Рассмотрим пример антипаттерна Chatty Service. Его подробное описание можно найти по адресу: <http://sofa.uqam.ca/resources/antipatterns.php>. По сути, антипаттерн Chatty Service (см. «Рис. 1») состоит из нескольких сервисов, которые обмениваются друг с другом большим количеством сообщений малого размера, чаще всего, примитивного типа.

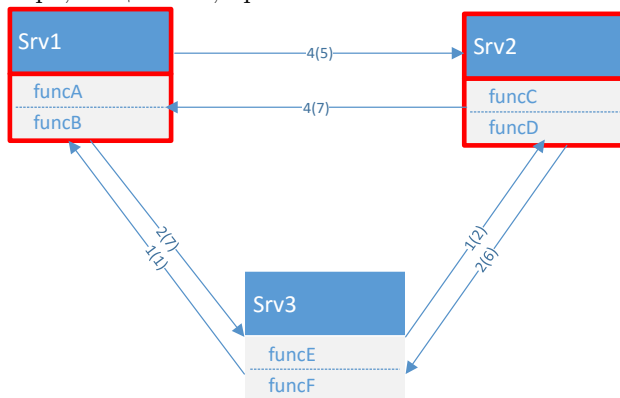


Рис. 1. Демонстрация антипаттерна «Chatty Service»

Таким образом, чтобы определить правило для определения антипаттерна Chatty Service, необходимо выделить следующее правило, состоящее из двух метрик и соответствующих им значений:

- (6) количество взаимных вызовов: высокий уровень;
- (7) связность с другими сервисами: высокий уровень. Высокий уровень соответствует показателю 80% и выше.

Для автоматизации процесса обнаружения антипаттернов был создан исследовательский прототип информационной системы, реализующей описанный подход.

## **Заключение**

Данная работа посвящена вопросу необходимости мониторинга состояния СО программных систем в условиях непрерывного развития и совершенствования, когда размер и сложность систем расширяется слишком быстро, чтобы человек мог справиться подручными средствами.

В контексте данной работы был произведен обзор методов для выявления антипаттернов а также сферы извлечения знаний из логов событий.

В ходе исследования были предложены правила обнаружения некоторых антипаттернов. Правила состоят из нескольких метрик и их конкретных значений, описывая симптомы антипаттернов. На текущий момент используется пять метрик: количество входящих и исходящих вызовов, время отклика, связность с другими сервисами. С применением этих метрик были предложены алгоритмы обнаружения нескольких антипаттернов, один из примеров приведен в статье.

Алгоритмы обнаружения антипаттернов были реализованы в виде исследовательского прототипа, который позволяет, задав правила в формате XML, обнаруживать антипаттерны.

В дальнейшем планируется увеличение количества вычисляемых метрик, и находимых с их помощью антипаттернов. Кроме того, планируется введение динамического отслеживания изменения значения метрик.

## **Список литературы**

- [1] Hu L., Sartipi K. Dynamic analysis and design pattern detection in java programs // Proceedings of SEKE'2008, 2008, pp. 842–846.

- [2] Kessentini M., Vaucher S., Sahraoui H. Deviance From Perfection is a Better Criterion Than Closeness To Evil When Identifying Risky Code // Proceedings of the IEEE/ACM ASE. ACM, 2010, pp. 113–122.
- [3] Lanza M., Marinescu R. Object-Oriented Metrics in Practice. - Springer-Verlag, 2006.
- [4] Moha N., Guéhéneuc Y.-G., Duchien L., Meur A.-F. L. DECOR: A Method for the Specification and Detection of Code and Design Smells // IEEE Trans. Software Eng. 36(1):20–36, 2010.
- [5] Nayrolles M., Moha N., Valtchev P. Improving SOA Antipatterns Detection in Service Based Systems by Mining Execution Traces // Proceedings of the 20th Working Conference on Reverse Engineering (WCRE'13), pp. 321–330, IEEE, 2013.
- [6] Nayrolles M., Palma F., Moha N., Guéhéneuc Y.-G. SODA: A Tool Support for the Detection of SOA Antipatterns // ICSOC Demonstration Track (Shanghai, China), in conjunction with ICSOC 2012, (10th International Conference on Service Oriented Computing), November 12-16, Shanghai, China, 2012.
- [7] Ng J. K.-Y., Guéhéneuc Y.-G., Antoniol G. Identification of behavioural and creational design motifs through dynamic analysis // JSMRP, vol. 22, no. 8, pp. 597–627, 2010.
- [8] F. Palma, L. An, F. Khomh, N. Moha, Y.-G. Guéhéneuc Investigating the Change-proneness of Service Patterns and Antipatterns // 7th IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2014), 17-19 November 2014, Matsue, Japan.
- [9] Palma F., Nayrolles M., Moha N., Guéhéneuc Y.-G., Baudry B., Jézéquel J.-M. SOA Antipatterns: an Approach for their Specification and Detection // International Journal of Cooperative Information Systems, 22 (04), 2013.
- [10] Settas D. L., Meditskos G., Stamelos I. G., Bassiliades N. SPARSE: A symptom-based antipattern retrieval knowledge-based system using Semantic Web technologies // ESA, vol. 38, no. 6, pp. 7633–7646, June 2011.
- [11] Upadhyaya B., Tang R., Zou Y. An approach for mining service composition patterns from execution logs // Journal of Software : Evolution and Process, 2012.
- [12] van der Aalst W.M.P. Process Mining: Discovery, Conformance and Enhancement of Business Processes. -- Berlin, Germany: Springer Verlag, Berlin-Heidelberg, 2011.

- [13] Wong S., Aaron M., Segall J., Lynch K., Mancoridis S. Reverse Engineering Utility Functions Using Genetic Programming to Detect Anomalous Behavior in Software // Proceedings of WCRE. IEEE Computer Society, 2010, pp. 141–149.

*Об авторе:*



**Александр Сергеевич Югов**

студент 2-го курса магистерской программы «Системная и программная инженерия» Национального исследовательского университета «Высшая Школа Экономики».

*e-mail:* yugovas@live.ru

*Образец ссылки на публикацию:*

А. С. Югов. О применении анализа журналов событий для обнаружения антипаттернов в сервис-ориентированных системах // Программные системы: теория и приложения: электрон. научн. журн. 2013. Т. 4, № 3(17), с. ??–??.

URL:

<http://psta.psiras.ru/read/???>

A.S. Yugov. On the application of the event log analysis to detection of antipatterns in service-based systems.

ABSTRACT. Like any other complex software system, Service Based Systems must evolve to fit new user requirements and execution contexts. The changes resulting from the evolution of SBSs may degrade their design and quality of service and may often cause the appearance of common poor solutions, called Antipatterns. Antipatterns resulting from these changes also affect the future maintenance and evolution of SBSs. The automatic detection of antipatterns is thus important to assess the design and quality of SBSs and make their maintenance and evolution easier.

This paper will focus on exploring rules to recognize symptoms of anti-patterns from data mined from software logs and to develop prototype of software system implementing these rules. аннотации на английский язык.

*Key Words and Phrases:* Services; Antipattern Detection; Process Mining.