An Approach for Service Usage Profiles Discovery*

Bruno Vollino¹, Karin Becker¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{bruno.vollino,karin.becker}@inf.ufrgs.br

Abstract. Specially in large scale services, changes in the provider can result in a different impact over distinct groups of clients, according to their service usage. Therefore, the provider must analyze the external integration perspective when planning changes, what refers to the compatibility of changes over clients. In this work we propose an approach to discover service usage profiles, based on data mining, through the clustering of client applications accordingly to its usage data. The proposed approach can help in service evolution management, specifically in the assessment of external impact of changes. Finally, we present the current status of the research and the future works.

Resumo. Especialmente em serviços web de larga escala, as mudanças feitas pelo provedor podem impactar de forma distinta seus diferentes grupos de clientes, conforme o uso. O provedor precisa, portanto, planejar mudanças sob a perspectiva da integração externa, o que significa avaliar a compatibilidade de mudanças em relação aos seus clientes. Este trabalho propõe uma abordagem para descobrir perfis de uso de serviços, baseada em mineração de dados, através do agrupamento de aplicações cliente de acordo com dados de uso. A abordagem proposta é capaz de auxiliar a geração da evolução de serviços, mais especificamente na avaliação do impacto externo de mudanças. Ao final, são apresentados o estado atual da pesquisa e os trabalhos futuros.

Palavras-chave: Web Service, Service Usage, Data Mining, Usage profile

*This research is financially supported by FAPERGS (Grant 11/1261-7), CNPq and CAPES - Brazil.
1. Introduction

Web services are software components designed to provide a standardized and platform independent manner to interoperate systems, which allows the development of loosely coupled applications. The adoption of service-oriented computing is highly motivated by its ability to deal with changes. But services themselves also undergo changes, and providers want to continue supporting existing customers, as well as attracting new clients. If not carefully planned, changes that are incompatible with current usage can break clients. Examples of disruptive changes are the non-backward incompatible changes or the decommission of service versions [Andrikopoulos et al. 2012].

Typically, the assessment of change impact is based on the worst case scenario, i.e. potential impact [Andrikopoulos et al. 2012] [Wang and Capretz 2009]. But, particularly in a large-scale usage scenario, services can be used in many different ways. This leverages a need for methods and tools that support providers in decisions of which and how changes will be executed, based on the understanding of how changes will affect the current clients. We have proposed the analysis of client usage profiles to measure the impact of changes [Yamashita et al. 2012] and to support decision in evolution management [Silva et al. 2012]. Client profiles represent significant consuming patterns in terms of service features (e.g. usage of a specific set of operations).

In this work, we present an approach to discover service usage profiles, by applying data mining techniques to cluster clients according to usage patterns. We discuss the knowledge discovery process over service interactions and the Profile Manager, a software component that supports this process. It handles the interception and logging of service requests, the creation of a clean and enriched Usage Database, the clustering of similar clients and their representation as profiles.

The analysis of clients usage patterns have been studied for other purposes such as workflow discovery and service recommendation [Motahari-Nezhad et al. 2011, Shi et al. 2011, Zhang et al. 2011, Tang and Zou 2010]. The assessment of change impact only in the service level, based on business process data, was also explored [Wang and Capretz 2009]. The proposed approach differs from other works by exploring usage patterns in the feature level (e.g. operations, data types), and by addressing specific issues of the change management problem, such as the adequate data preparation and profile representation according to the management task. The use of usage data to cluster clients is also a differential, since that clients composition information is frequently unknown in large-scale services.

The remaining of this paper is structured as follows. Section 2 discusses the related work. Section 3 describes in detail the proposed approach to discover service usage profiles. Section 4 draws conclusions and presents future work.

2. Related Work

There exists a wide range of applications for usage mining in services. Examples are the prediction of development costs, performance monitoring, and services recommendation [Nayak 2008]. We include in this list the prediction of business costs of changes [Silva et al. 2012], quantification of external integration impact (compatibility of changes over clients) and identification of evolutive usage trends [Yamashita et al. 2012].
The challenges in service mining are many. Experiments are frequently executed over synthetic data [Tang and Zou 2010, Zhang et al. 2011], due to the proprietary characteristics of service usage data [Nayak 2008]. The evaluation and validation of results is also difficult, due to the volume of data and the patterns complexity. The interception of service interactions is another issue, due to the distributed nature of services. The trade-offs of a diversity of monitors have been studied [Chuvakin and Peterson 2009].

Examples of usage mining applied to service recommendation include the discovery of similar users based on query-invocation relationships of a service registry [Zhang et al. 2011] and the clustering of clients based on their location and QoS statistics [Shi et al. 2011]. Other usage mining applications are the discovery of workflows in usage data [Motahari-Nezhad et al. 2011] and composition patterns [Tang and Zou 2010], and the quantification of change impact, as example, by clustering services based on services dependency patterns within business process descriptions [Wang and Capretz 2009]. To the best of our knowledge, there is no work similar to our proposal. We discover client clusters instead of workflows or composition patterns, and our cluster definition is based on usage, instead of query-invocation, QoS similarity or process information.

3. Profile Discovery

Service usage profiles are abstract representations of groups of client applications with similar usage patterns. Given our focus is to assess the impact of changes in service evolution, we address usage patterns as the use of features of a specific service interface version by client applications. The term feature comes from the underlying service representation model we adopt [Yamashita et al. 2012], which divides a service description into smaller fragments related to the service as a whole, an operation description, or message/data type description. The profile structure is depicted in Fig. 1. It associates each profile with the features used and related metrics (e.g. number of requests), as well as with specific features of that service version and their related metrics (e.g. number of requests of an operation or the frequency with which a type was exchanged in messages).

![Figure 1. Usage Profile structure](image)

The logging of all requests to a service version generates a huge set of complex usage data. To find patterns in this data, we execute a knowledge discovery process, using clustering as the data mining technique [Han and Kamber 2006]. We have designed the Profile Manager software component to support this process, which is part of a change management framework [Yamashita et al. 2012]. It is integrated with a service version manager, from which it extracts meta-information about service structure and versioning. The Profile Manager, detailed in the remaining of this section, is shown in Figure 2.

3.1. Interaction Monitor

This component is responsible for intercepting messages exchanged between client applications and the service versions, and logging them into interaction logs. There
are several alternatives for the monitoring infrastructure, each one imposing trade-offs in terms of scope of extractable data and performance of the monitoring capabilities [Chuvakin and Peterson 2009].

In our architecture, the Interaction Monitor resides in the server side, since a client side approach requires changes to the client and frequent consolidation of distributed data. We assume that the service has an authentication mechanism, to be able to associate interactions with its respective clients. For each client request or service response, we extract and log information about the features used: the service version, the operation called and the parameters exchanged. Notice that parameters usage influences the generated profiles, meaning applications usage patterns are compound of operations and the data types they exchange through messages. By now we cover only SOAP based services. We log the SOAP messages in a suitable XML file, by the use of a message handler.

3.2. Data Loader

The Data Loader extracts the raw data from the interaction logs of the different versions of a service, cleans and transforms it, and loads the processed data in a Usage Database, as a graph, of which the schema is depicted in Fig. 3.

Only the information about the services interactions and the types of parameters involved in the interaction are necessary, and thus their actual values are disregarded. The loader handles noise in the logs by removing incomplete log entries (e.g. failures), as well as requests that make an improper usage of the service (e.g. missing arguments). Finally, the loader enriches this data with web service versions structure as available in the Version Manager. The service, operation and parameters used in an interaction constitute a hierarchical structure. We use a graph database management system to store the service interactions and its structures efficiently.

The Usage Database facilitates the process of preparing data for accomplishing different analysis purposes. For instance, we can select the client interactions together with used features in three levels (service, operation and type), enabling the definition of usage profiles with distinct usage granularities.
3.3. Profile Generation

The Profile Generator is responsible for clustering applications based on similar feature usage, and after post-processing the resulting clusters, creating service usage profiles. It extracts the relevant usage data from the database, and transform it for the application of the clustering algorithm.

Given our interest on the evaluation of changes between two versions, we extract data corresponding to the features of a single service version. We also filter usage data collected in a certain time interval, because usage patterns may change over time.

This data is aggregated and transformed in a tabular representation, where rows represent unique clients, and columns represents feature versions. Each row is thus an aggregation of all interactions of a same client with regard to the features. To represent the usage of features, one can choose between a binary or weighted approach. The latter represents the quantification of requests for that feature. The choice depends on the analysis requirements of the change management task (e.g. usage oriented compatibility, change impact assessment). Other standard preparation tasks can be applied for improving the results of clustering (e.g. normalization, outliers) [Han and Kamber 2006].

Then, clustering is applied over the prepared data. Various clustering algorithms exist, which handle similarity differently [Han and Kamber 2006], and require distinct parameters. The number of clusters must be inferred from usage data by the algorithm or in a pre-clustering step. Web services contain, in general, processes or workflows [Motahari-Nezhad et al. 2011], which are probably the most frequent patterns observed in the usage data. Distinct workflows must be well separated by the resulting clusters, perhaps the alternative flows must be not ignored. Thus, the algorithm must be low susceptible to noise, but sensible to lesser frequent workflow patterns.

Once clusters are generated, their validity needs to be evaluated. Options are internal criteria such as inter and intra cluster distances, and external criteria (e.g. error measures for classification). Due to the lack of real data, we chose to evaluate the approach with a synthetic dataset containing predefined usage profiles, and use an external measure to assess clusters’ validity.

The Profile Manager finally post-processes the discovered clusters to define usage profiles, which are representative summarizations of each cluster of client applications. Metrics can be associated to profiles or to the features used, which are extracted from the Usage Database (e.g. requests). The quality of the generated profiles can be evaluated by verifying how well they summarize usage data for the analysis purpose, e.g. compare impact calculated with profiles against impact calculated with the complete dataset.

4. Conclusions and Future Work

This paper presented an approach for discovering service usage profiles based on usage data. The Profile Manager supports that process, which is integrated in a broader change management framework [Yamashita et al. 2012, Silva et al. 2012]. We are currently implementing the Profile Manager and testing data mining alternatives with synthetic usage datasets. We also plan to develop a semi-automatic validation method for usage profiles, so one can assess the validity before persisting results.

The discovery of profiles is very sensitive to the choices for preprocessing data,
clustering and validating clusters. Clusters are dependent on how usage is characterized in the dataset: binary, weighted by use or weighted by usage relevance. The choice of either approach is based on the analysis requirements of the management task being executed. By selecting appropriate client representations and clustering algorithms, we plan to cover the tasks of usage oriented compatibility, change impact assessment and analysis of usage trends (profile evolution). So far, we obtained best results with density-based clustering algorithms, on small datasets. We are developing more elaborated experiments to select the best techniques and client representation for each task.

The proposed approach is targeted at the assessment of changes such that the provider could decide about changes implementation according to their effect on clients, but this solution could also be leveraged for performance analysis (e.g. overload balancing and message redirection), or for service recommendation.

References


