

StakeCloud Tool: From Cloud Consumers' Search Queries to New Service Requirements

Irina Todoran Koitz, Martin Glinz

Department of Informatics

University of Zurich, Switzerland

{koitz, glinz}@ifi.uzh.ch

Abstract—Requirements elicitation is indispensable for delivering successful services. Nevertheless, cloud service providers mostly rely on ad-hoc approaches, as there are no dedicated elicitation methods for cloud services. To address this problem, we developed the StakeCloud approach, which helps cloud providers elicit requirements for future cloud services. StakeCloud builds and analyzes fuzzy Galois lattices based on consumers' advanced search queries for cloud services. Our StakeCloud Tool automatically builds the lattice from the given search queries. It provides the requirements analyst with extensive clustering and analysis capabilities as well as means for comparing different newly generated classes of services. These allow identifying the threshold for achieving the largest populations of satisfied consumers with a minimum set of features implemented. Further, our tool enables eliciting real requirements from global consumers unobtrusively.

I. INTRODUCTION

Understanding consumers' requirements is critical for building successful services [1]. Therefore, a wide range of requirements elicitation methods has been developed and used in practice over the last decades. However, in the domain of cloud services, the existing techniques are challenged [2]. The cloud is not only bringing technical and economic benefits [3], but also difficulties regarding requirements elicitation, e.g., heterogeneous and globally distributed consumers. According to our previous work [2], the most popular requirements elicitation techniques among cloud providers are traditional methods (questionnaires, surveys, interviews, document analysis) and prototyping. Nevertheless, these are inadequate for the cloud: e.g., they require long elicitation times, can rarely be applied remotely and do not support automation. Hence cloud service providers use ad-hoc approaches such as inventing, guessing requirements or imitating competitors, which lead to dissatisfaction or failure-prone services.

To address this issue, we introduced the StakeCloud approach for eliciting cloud service requirements [4]. With the emergence of marketplaces as search platforms for cloud services (e.g., Intel Cloud Finder), cloud consumers have the opportunity to input their advanced search queries for cloud services to find solutions that match their needs. Based on such queries represented by finite sets of feature values collected on marketplaces, StakeCloud models fuzzy Galois lattices which are then analyzed to infer new cloud service requirements. Lattices are represented graphically as acyclic directed graphs having exactly one source node with no incoming edges and one sink node with no outgoing edges (for an example,

see Fig. 2). In lattices, any two nodes have at least one supremum and one infimum element. For more information on our theoretical approach, please refer to [4]. In this tool demo paper, we describe the StakeCloud Tool and how it can be used by cloud providers in their decision-making process. Our main contribution is a novel tool-supported requirements elicitation technique tailored to a cloud computing context.

II. THE STAKECLOUD TOOL

Starting from our theoretical solution [4], we developed the StakeCloud Tool, which automates the requirements elicitation activity for cloud services to a large extent, as shown in Fig. 1. Once the queries are collected (A1), the tool checks for duplicates (A2) and counts the individual frequencies (A3). Then, the cloud provider analyst models the queries as fuzzy vectors (A4: this activity is partially automated by the tool) and decides whether (s)he wants to cluster similar queries (A5). If applicable, the tool computes the clusters (A6) and then generates the Galois lattice automatically (A7). At this stage, the cloud provider can iteratively choose from and apply the available analysis criteria (A8). The tool will automatically generate the corresponding graphs (A9), indicating what combinations of features and feature values satisfy the largest number of input consumer queries with a minimum compromise. These represent draft requirements for new services that the cloud provider representative can use in his/her decision-making process regarding upcoming service releases. Naturally, our approach and tool can be complemented by existing more in-depth requirements elicitation techniques.

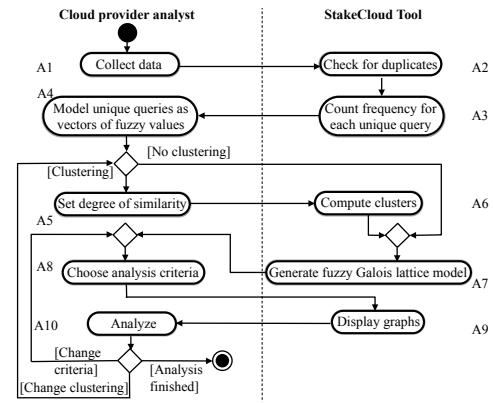


Fig. 1. UML activity diagram of the process of our approach.

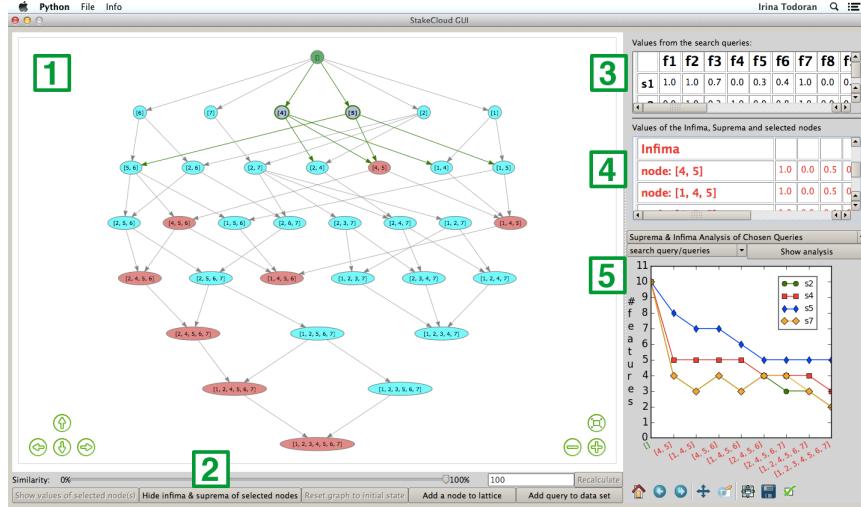


Fig. 2. StakeCloud Tool screenshot.

Figure 2 presents a screenshot of the StakeCloud Tool, where the main aspects are indicated with green numbers (1-5). Once an input file containing consumers' advanced search queries (encoded as fuzzy vectors) is loaded, the corresponding fuzzy Galois lattice is *automatically generated* and *displayed* in the main panel of the window (1). The elements in the first level of the lattice hierarchy, i.e., nodes identified by only one digit, represent the initial search queries, whereas the other elements represent new classes of potential cloud services generated by our approach. The user (cloud provider) can *select* (e.g., nodes [4] and [5]) and *drag* nodes, choose to *display the actual fuzzy values* composing the selected nodes or *highlight the suprema and infima elements* of particular nodes by clicking the buttons in the lower part of the window (2). The supremum is a comprehensive service that satisfies all the features requested in the corresponding subset of queries. However, this is most often impossible or too expensive to implement. Hence the infimum elements are better candidates for development, since they fully satisfy only those features that all queries in the corresponding subset have in common, while the remaining ones are partially or not satisfied. Our tool provides compromise analysis support regarding the implementation of requirements for future services.

If the cloud provider analyst is interested in seeing where his/her offering is situated in the lattice among the queries and generated classes of services, (s)he can choose to *add a node to the lattice* (2) by typing the feature values for his/her service. Moreover, in case the lattice is too large for a good visualization, the *zooming* functionalities or the *similarity degree* can be used. By setting a similarity degree (2), similar queries are *clustered*, thus reducing the overall lattice size. However, the composing queries can still be analyzed and viewed in detail. In the right side of the window, the top panel (3) *shows the values of the input search queries*, and the middle panel (4) displays the fuzzy vectors for *selected nodes* or *infima* (red) and *suprema nodes* (green). The bottom panel (5) exhibits five *lattice analysis options* in a drop-down menu, e.g., analyze to what extent individual features are accomplished by the

selected nodes, or how many features are fully satisfied by their suprema/infima. The video at <http://goo.gl/qv5I5s> shows a practical scenario of the StakeCloud Tool.

As far as the technical implementation is concerned, our tool was developed in Python 2.7, using PyQt4 for the graphical user interface. Since PyQt is a Qt binding for Python, the StakeCloud tool is cross-platform and has been tested in Mac OS 10.x, Windows 8 and Linux Ubuntu 14.10. For visualizing the lattice, the vis.js library was used, whereas the analysis graphs are matplotlib graphs.

III. CONCLUSION AND FUTURE WORK

The StakeCloud Tool implements the fuzzy Galois lattices approach for requirements elicitation in cloud settings. It allows analyzing advanced search queries for cloud services to infer new requirements, thus semi-automating the elicitation activity. This way, our approach and tool enable large-scale consumers' involvement and is best suited for the early elicitation phase and for monitoring market trends. So far, the technical performance of the StakeCloud tool has been evaluated with self-generated datasets, yielding promising results, and we plan to further use the tool with cloud provider companies.

ACKNOWLEDGMENT

The authors would like to thank Matthias Scherrer for his contribution. This work is supported by the Swiss National Science Foundation (SNSF), grant no. 200021_150142.

REFERENCES

- [1] I. Sommerville and P. Sawyer, *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.
- [2] I. Todoran, N. Seyff, and M. Glinz, "How cloud providers elicit consumer requirements: An exploratory study of nineteen companies," in *21st IEEE Intl Requirements Eng Conf (RE'13)*, 2013, pp. 105–114.
- [3] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the cloud? An architectural map of the cloud landscape," in *ICSE Workshop on Software Engineering Challenges of Cloud Computing*. IEEE Computer Society, 2009, pp. 23–31.
- [4] I. Todoran and M. Glinz, "Quest for requirements: Scrutinizing advanced search queries for cloud services with fuzzy Galois lattices," in *IEEE 10th World Congress on Services (SERVICES)*. IEEE, 2014, pp. 234–241.