RESDEC: Online Management Tool for Implementation Components Selection in Software Product Lines Using Recommender Systems

Jorge Rodas-Silva
University of Milagro
Milagro, Ecuador
jrodass@unemi.edu.ec

José A. Galindo
Jorge García-Gutiérrez
David Benavides
University of Seville
Seville, Spain
{jagalindo,benavides,jorgarcia}@us.es

ABSTRACT

Software product lines (SPL) management is one of the most important activities for the software engineer and it represents one of the key pieces of software product line engineering. When a software system grow fast, configuring a product becomes a costly and error-prone activity due to the amount of features available for configuration. This process becomes more complex when for each feature, there is more than one component that implements it. Currently the tools available for configuration management do not have automated mechanisms to facilitate the optimal components selection that meet the functions required by a given product. In this paper, we introduce a prototype component-based recommender system called RESDEC (REcommender System that suggest implementation Components from selecteD fEatures) designed to manage the best implementation components alternatives. Our tool is validated using WordPress-based websites where the implementation components are represented by plugins and the recommendations generated by RESDEC help interested parties in the search and efficient plugins selection to configure websites.

CCS CONCEPTS

• **Software and its engineering → Software development pro-cess management**; • **Software product lines**;

KEYWORDS:

implementation components,
software product lines,
recommender systems,
WordPress

1 INTRODUCTION

Although the process of creating new software has benefited from the functionalities availability in the form of components, obtaining an adequate configuration that meets a set of specific requirements is a complex activity. This is mainly due to the large number of options (specific application variants), which can be generated from the existing components combination. In an SPL, feature models capture the common and variable aspects of a family of similar products [2], which allow the generation of different software variants.

A common example of a software variant is a website designed in WordPress. Normally, a software developer manually searches for those components (plugins) that are feasible and most optimal for each website. This task takes time and does not always guarantee the selected components to be the most suitable (in terms of quality) for the required application. Two scenarios could arise during this configuration, on one hand, empirically selecting a component, in the practice, may not provide the expected results; and on the other hand, not having the criteria based on other users' experience regarding these components, could induce a bad selection and achieve a bad experience for the end user.

There are several tools in the literature to support the SPL configuration process [3, 10–14]. However, to the best of our knowledge and understanding, none of the tools developed so far supports the selection of implementation components when configure a product.

To overcome this problem, in this paper we introduce a component-based recommender system tool called RESDEC whose main focus is the use of explicit information that is generated by users about the implementation components and that is stored in an information repository.

The information provided by the users allows RESDEC to generate recommendations in real time through the use of recommender systems based on collaborative filtering and content filtering. In the practice, there are successful experiences with recommender systems, among which, online stores (Amazon), providers of state-of-the-art services (Netflix), among others stand out [1]. Taking the advantages offered by these systems in the context of selecting components to configure and customize software product lines is one of the main objectives pursued by RESDEC.

RESDEC supports a set of recommender algorithms based on the method described in [15] for the components selection. Our tool uses these algorithms in three different scenarios which originate

during the configuration of a website in WordPress: i) *Cold start*, which is executed when there is no information associated with the user, that is, when a user is going to set up a website for the first time. That is, this scenario recommends a list of plugins based on the trend or popularity; ii) *Recommendations of implementation components based on ratings*, which recommends plugins based on the experience of users with similar profiles according to the ratings; and iii) *Recommendations of implementation components based on features*, which recommends plugins based on contextual information about them, that is, the tags associated with the plugins used by users in the past.

2 RESDEC GENERAL OVERVIEW

RESDEC supplies users with information about which components are more suitable to use for SPL configuration. Next, we present the requirements and main elements of our tool.

2.1 Requirements

When managing an SPL of based-websites on WordPress, there is a large number of implementation components associated to features, such as payment options, shopping cart, security controls, among others. With a large number of features, managing the configuration of a web site on WordPress becomes a difficult task, even more difficult when, for each feature, there is more than one possibility of implementation. Therefore, implementing the website with a combination of appropriate components can be a complex activity to solve.

RESDEC uses feature models to describe the set of valid configurations. From these models, we can identify the implementation components associated with each feature and obtain the information that is developed around them. The processes for building these feature models are beyond the scope of this paper.

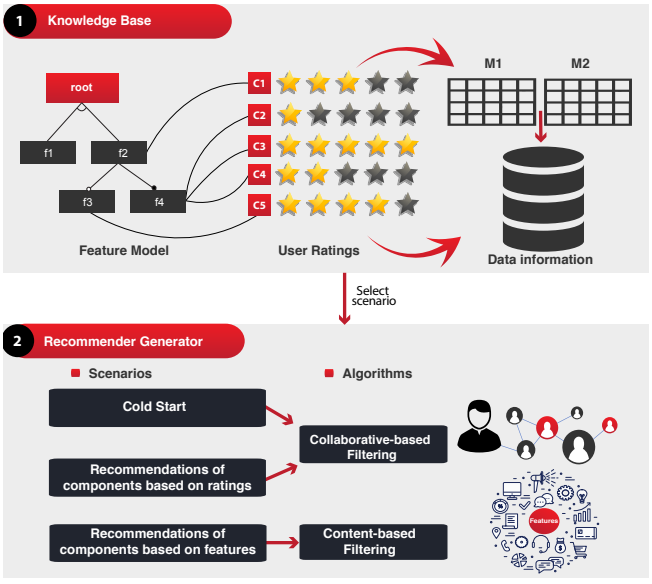

Figure 1: RESDEC elements

2.2 Elements

Fig. 1 illustrates the basic elements of our tool. As a first element, we have the *feature model* that describes the variability present in an SPL domain. From the feature model, it is possible to select a set of features that are associated to a series of components. Some of these implementation components contain information that is generated by users, such as, components ratings, report of bugs, number of installations, test cases executed, etc. In our case, we considered ratings that users make on WordPress plugins that have been used in website configuration in the past. The mapping of the feature model to the implementation components are described on the RESDEC website[1].

The information collected from the users allowed us to build the *Knowledge Base* (KB) composed of two matrices, $M1$ and $M2$. Matrix $M1$ relates user information, implementation components and ratings; while the matrix $M2$ relates the information of implementation components with their associated features.

After the knowledge base is built, we run the *Recommenders Generator*, which uses a set of algorithms that are commonly used in a Recommender Systems [4]. Specifically, we use collaborative-based and content-based recommender algorithms. Collaborative-based recommender systems [5, 7], also known as personalized recommender systems, are based on the analysis of user profiles, where recommendations are generated according to the tastes of users with similar preferences; while, content-based recommender systems [8] make recommendations based on the characteristics of the items without need to use information from other users.

These algorithms we have implemented in three different scenarios: *(i)* cold start; *(ii)* recommendations of implementation components based on ratings; and *(iii)* recommendations of implementation components based on features (see section 3.3).

For first scenario, we have implemented a classical popularity algorithm. While for the algorithms which run in the second scenario, we have employed Item-item KNN [16], User-user KNN [6] and the SVD factorization matrix [16]. Finally, for third scenario, we have implemented the algorithm TF-IDF [9].

In either case, regardless of the scenario and the selected algorithm, RESDEC recommends to the user a list of implementation components that could be used to configure a product in an SPL.

3 RESDEC TOOL SUITE

RESDEC offers two main functionalities: component repository management and automated analysis in the implementation components selection through recommender systems. The following are some advantages of RESDEC:

- It is easy adapt to any SPL configuration environment. To do this, the knowledge base has been designed based on three attributes commonly used by a recommender system (i.e users, items and ratings). This allowed us, that algorithms implemented in RESDEC receive these parameters as input and run without problems in any SPL scenario, for example: WordPress, Android, Mozilla, among others.
- It offers information about the implementation components and the ratings history made by the user.

[1]Real Case applied an eCommerce Site: http://resdec.com/about-case-study.html

– It provides a set of recommender algorithms that can be extended to provide better recommendation results in the three scenarios presented in this paper.
– It offers on screen, an updated history of the last components of implementation that have been of interest to the user.
– It allows obtaining recommendations, in execution time, of the most appropriate implementation components according to the feature selected by the user.
– It incorporates a case study based on a website software product lines that validates the scope of our tool.

3.1 Dataset

The dataset used by RESDEC has been built from CSV files. The data collected by these files corresponds to information extracted from WordPress. To obtain the data we have created a selenium based crawler[2]. First, we extract the list of plugins from WordPress[3], then, through the different plugins obtaining its number of stars, downloads, version, last update date, the WordPress version, the required PHP version, and associated tags. Finally, we obtain the list of users that reviewed the plugin, among its concrete review, and score. This information is then stored in a Json file[4] which is later exploited to generate the required inputs for RESDEC.

3.2 Architecture

RESDEC Tool has 3 components: a repository manager, a recommender manager and an output manager.

The *repository manager* responds to the requests of the stakeholders and structures the matrices $M1$ and $M2$ of the *DBKnowledge* (given in 2.2) through CSV's.

The *recommender manager* is in charge of processing the recommendations. It is developed in *Python* with a package of libraries that contain the algorithms that the recommender manager runs according to the scenario selected by the stakeholder.

For the *Cold Start* scenario RESDEC uses a classical popularity algorithm. While for the algorithms that run in the scenario *Recommendations of implementation components based on ratings*, employs the Scikit-surprise library[5]; and for the *Recommendations of implementation components based on features* scenario, it uses the Scikit-learn library[6].

The recommender manager is scalable and offers the possibility of implementing new recommender algorithms in any of the three scenarios presented in this paper.

The *output manager* interacts directly with the stakeholder using the repository manager and the recommender manager to generate the list of recommendations of the implementation components. It is designed in HTML5 and JavaScript, supported by the Semantic UI framework[7] used for the design of the interfaces. The interaction between the stakeholder and RESDEC is done through a web browser.

[2]Selenium WebSite: https://www.seleniumhq.org/
[3]WordPress plugins: https://wordpress.org/plugins/browse/popular/
[4]Json WebSite: https://www.json.org/
[5]Surprise website: http://surprise.readthedocs.io/en/stable/
[6]Scikit-learn website: http://scikit-learn.org/stable/index.html
[7]Semantic UI website: https://semantic-ui.com/

3.3 Web Application

To make our work accessible to the community, we present a RESDEC web application that eases the generation of recommendations to stakeholders that require guided assistance in the selection of plugins to configure a products line based on WordPress websites (see Fig. 2).

The main screen of RESDEC presents a menu with three recommendation scenarios where stackholders or users can configure an SPL, additionally incorporates a case study about eCommerce websites developed in WordPress (see Fig. **??** in Appendix B). RESDEC is available at *www.resdec.com*.

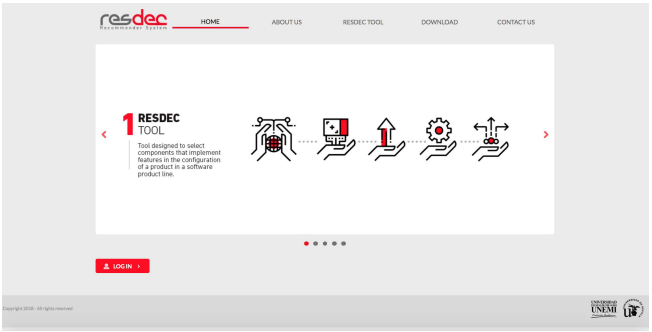

Figure 2: RESDEC web application

Next, we describe the different functionalities of our tool through an example based on the configuration of a tourism website in WordPress.

(1) The *Cold Start* option recommends components when there is no information associated with the user profile, i.e., when the user has no experience and is setting up a website for the first time. Suppose we are going to set up a new tourism website and we need to implement the *Social Media* function. In this case, the user selects the tag or tags associated with *Social Media* and specifies the number of desired recommendations. With the information provided by the user, RESDEC sets the recommendations based on the popularity of WordPress plugins and does not use the information associated with the user's profile (see Fig. **??** in Appendix B).
(2) The second option, *Recommendations of implementation components based on ratings*, from a component used by the user in previous configurations, recommends those components that users with similar profiles have used in the configuration of a product. Suppose that we are going to configure a *tourism website* that has already been implemented and in which we need new recommendation alternatives for *Social Media* function. In this case, the user selects the implemented plugin, *social-media-widget*, then specifies the number of desired recommendations and selects the algorithm to execute (SVD, item-item KNN or user-user KNN). With the information provided by the user, RESDEC establishes the recommendations based on the ratings that plugins similar to the one selected have been used by other users. In this scenario, RESDEC uses the information associated with the user's profile (see Fig. **??** in Appendix B).

(3) The third option, *Recommendations of implementation components based on features*, recommends implementation components based on the features of a component used by the user in previous configurations. That is, the list of recommendations is established based on the descriptive information of the components associated with the user profile. Suppose that we are going to configure a *tourism website* that has already been implemented and we need to replace or complement the feature *Social Media*, in this case, first the user selects the plugin implemented, *social-media-widget*, then the system will display the list of tags associated with the plugin through which it will establish recommendations. Then, we specify the number of desired recommendations and select the execution algorithm (TF-IDF). In this case, RESDEC establishes the recommendations based on the features, that is, on the similarity of the tags associated to the selected plugin with other plugins that use one or more of these tags. In this scenario, RESDEC also uses the information associated to the user's profile (see Fig. **??** in Appendix B).

(4) The option *Case Study applied to eCommerce Website*, shows a Feature Model that was built from information about websites designed in WordPress that are available on the Internet. This Feature Model has been implemented in an interactive way and describes the relations between features, the same ones that can appear when configuring an eCommerce website on this platform.
In the Feature Model, for example, by clicking on the *Shopping Cart* feature, the lower part of the model is configured for each scenario, showing only information associated with the selected feature. Thus, for scenario 1 it will show only the list of tags associated to that feature, in the same way in scenarios 2 and 3, it will display only the plugins that implement that feature. The recommendations in each scenario are executed in a similar way as described above (see Fig. **??** in Appendix B).

Along with the list of recommendations shown in the options described above, RESDEC shows a tab called *"You might also be interested"* that shows other components that might be of interest to the interest group. Finally, there is an option called *"About this case study"*, when you click on this option, it shows in detail the process that was carried out to build the feature model.

4 CONCLUSIONS

We have described in this paper a tool called RESDEC that uses a recommender system for the selection of implementation components that helps stakeholders or user to configure the features of an SPL.

The tool is based on a set of collaborative filtering and content filtering algorithms that are commonly used in recommender systems and that are executed in three possible scenarios that may arise when configuring an SPL. To demonstrate the scope of our tool, we have used real information extracted from WordPress (users, plugins, ratings and tags) that has allowed us to make a case study of the possible problems that a web developer may face when configuring a website on this platform. Specifically, we focus on recommender to users plugins that allow a website to be configured in an application domain. The future work focuses especially on the implementation of new recommender algorithms, experimentation and adaptation to other scenarios and contexts of practical relevance.

REFERENCES

[1] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013), 109–132.

[2] José A. Galindo, David Benavides, Pablo Trinidad, Antonio Manuel Gutiérrez-Fernández, and Antonio Ruiz-Cortés. 2019. Automated analysis of feature models: Quo vadis? *Computing* 101, 5 (2019), 387–433. https://doi.org/10.1007/s00607-018-0646-1

[3] José A Galindo, Hamilton Turner, David Benavides, and Jules White. 2014. Testing variability-intensive systems using automated analysis: an application to Android. *Software Quality Journal* (2014), 1–41.

[4] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

[5] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM* 40, 3 (1997), 77–87.

[6] Shyong K Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*. ACM, 393–402.

[7] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE* 7, 1 (2003), 76–80.

[8] Michael Pazzani and Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine learning* 27, 3 (1997), 313–331.

[9] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[10] Juliana Alves Pereira, Kattiana Constantino, and Eduardo Figueiredo. 2015. A systematic literature review of software product line management tools. In *International Conference on Software Reuse*. Springer, 73–89.

[11] Juliana Alves Pereira, Jabier Martinez, Hari Kumar Gurudu, Sebastian Krieter, and Gunter Saake. 2018. Visual Guidance for Product Line Configuration Using Recommendations and Non-Functional Properties. (2018).

[12] Juliana Alves Pereira, Pawel Matuszyk, Sebastian Krieter, Myra Spiliopoulou, and Gunter Saake. 2018. Personalized recommender systems for product-line configuration processes. *Computer Languages, Systems & Structures* (2018).

[13] Juliana Alves Pereira, Sandro Schulze, Eduardo Figueiredo, and Gunter Saake. 2018. N-dimensional Tensor Factorization for Self-configuration of Software Product Lines at Runtime. In *Proceedings of the 22Nd International Systems and Software Product Line Conference - Volume 1 (SPLC '18)*. ACM, New York, NY, USA, 87–97. https://doi.org/10.1145/3233027.3233039

[14] Juliana Alves Pereira, Sandro Schulze, Sebastian Krieter, Márcio Ribeiro, and Gunter Saake. 2018. A Context-Aware Recommender System for Extended Software Product Line Configurations. In *Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems*. ACM, 97–104.

[15] J. Rodas-Silva, J. A. Galindo, J. García-Gutiérrez, and D. Benavides. 2019. Selection of software product line implementation components using recommender systems: An application to Wordpress. *IEEE Access* (2019), 1–1. https://doi.org/10.1109/ACCESS.2019.2918469

[16] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.