Automatic generation of personalized applications based on social media

Nadia Chouchani, Mourad Abed

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 825–830

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on the Advancements in Model Driven Engineering (AMDE 2020)
April 6 - 9, 2020, Warsaw, Poland

Automatic generation of personalized applications based on social media

Nadia Chouchani[a,*], Mourad Abed[b]

[a]IRT Railenium, Technople Transalley 180, rue Joseph Louis Lagrange, 59300 Valenciennes, France
[b]LAMIH UMR CNRS 8201, Le Mont-Houy, 59300 Valenciennes, France

Abstract

In this work, we present a contextual personalization method based on users' interests and social neighborhoods. It is implemented by proposing an approach to automatically generate personalized interactive applications based on social media platforms. The shared contents, designed by documents, are used to extract users' contexts and profiles. Thus, extensive functionality is needed to exploit these information and develop social applications adapted to end-users interests. An automated model-driven engineering approach, using Domain-Specific Language, is proposed for the generation of what we call personalized document-based applications by exploiting the interaction power of online social media.

Keywords: Model-Driven Engineering ; Domain-Specific Language ; Social media ; Personalization

1. Introduction

In order to take advantage of the success of social media platforms, many people are trying to implement social applications. They are based on interactions with these platforms and exploit their infrastructure as a means of dialogue with end-users. Adapting interactive applications to users' preferences and interests is a key factor in their success. The idea is to conquer them by providing relevant information using personalized systems. In this context, we propose an automated model-driven engineering (*MDE*) approach for the generation of interactive social applications with personalized content. The *MDE* methodology is an important paradigm given its ability to accelerate the development process and reduce its complexity. In order to meet this objective, we have identified the following requirements: (*i*) implementation of the personalization of content ; (*i*) taking into account this personalization from the earliest stages of the design of interactive applications ; and (*i*) automatic generation of the final application. The remainder of the

[*] Corresponding author.
E-mail address: nadia.chouchani@railenium.eu

10.1016/j.procs.2020.03.149

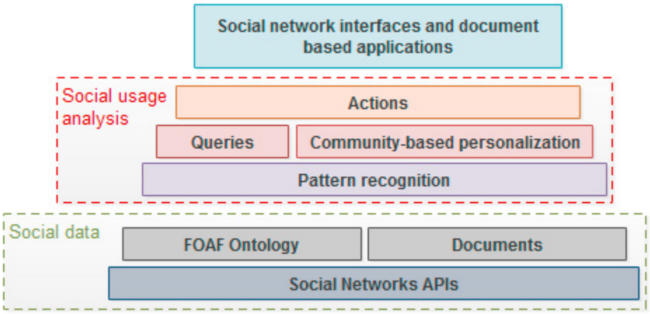


Fig. 1: Abstraction stack for the generation of interactive applications based on social media

paper is a follows. In section *2*, we present the general proposed solution. In section *3*, we detail the *MDE* approach. The section *4* presents the implementation and evaluation of the results. Finally, we conclude and describe future work.

2. General proposed solution

2.1. Collaborative contextualization method

Personalization is defined as the ability to provide users with content and services that are adapted to their needs and expectations at all times by using appropriate human-machine interactions [1]. In our work, we proceed with a personalization of content consisting in selecting and adapting relevant contextual information. The context corresponds to different dimensions, in particular the user dimension which includes the social and personal contexts [2]. The first is the social neighborhood of the user such as his groups and communities of interest. While the second contains its demographic, psychological and cognitive information. Thus, for the implementation of the personalization according to the users interests in addition to the social neighborhood, we will answer the following questions: (*i*) how to build the context? and (*ii*) how to use it to select relevant content?

The social user profile is a model that contains both contexts of the user dimension [3]. In fact, it is built not only on the basis of the interests and activities of the users but also taking into account other users who are similar to them. More specifically, and for better personalization of content, some research has explored the utility of exploiting group membership information that could be sufficiently similar to each other. This similarity is well guaranteed in communities of interest as user groups with common interests [4]. Thus, we propose to exploit the data in these communities by taking advantage of the common points of its members.

Definition (**Community-based Personalization**) : is a method of personalizing collaborative content considering the user as a member of a community of interest in order to provide him with information adapted to his interests.

For those purposes, we rely on the “groupization” technique [5]. It has been proven that this technique increases the personalization process. The latter is enhanced by assigning higher weights to documents of interest to more members of the group, based on matching the history of each member and the local frequencies of the documents. First, a personalization score is calculated for each search result for each member of the group. As for this calculating, we use the method defined by [6]. In their work, they consider a user’s documents such as web pages, email messages. In our work, we consider documents as textual content that the user publishes and shares in social media sites.

2.2. Abstraction stack for generating personalized applications

Figure 1 shows the stack of tools we have designed to generate personalized interactive applications based on social media. Its purpose is the integration the community-based personalization method into the design and subsequent generation of interactive applications. This stack is composed of: (*i*) user representation tools and extraction of doc-

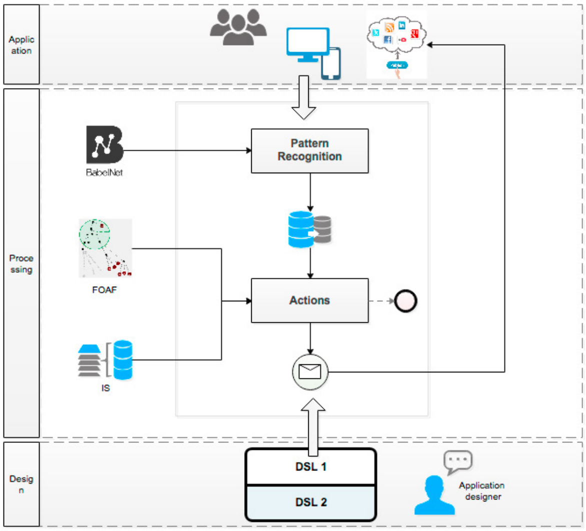


Fig. 2: General architecture

uments for the application through social media platforms; and (*ii*) querying and personalization tools based on the communities of interest in addition to the execution of actions on documents.

Social user profiles are represented by Friend Of A Friend (*FOAF*) ontology [3]. It is used to describe users, their communities and their contents. Social data are therefore available in a semantic format and can be exploited directly or interrogated using the *SPARQL*[1] language. Most of these data are accessible from the programming interfaces (*APIs*) of social media platforms. However, it is in our interest to analyze the available data in order to extract only the relevant information (pattern recognition *DSL*). Using the relevant documents and user queries for the application, we apply the community-based personalization method to select the relevant results that are adapted to the users' interests (actions *DSL*). The composition of the results is carried out by applying actions on the documents also allowing to send them back to the users through the *APIs*.

3. Proposed *MDE* approach

In this section, we present the architecture of the proposed approach for the automatic generation of personalized document-based applications. This architecture is organized in three layers (*figure* 2): an "application" layer represents the interface from which the user interacts with the application and a "processing" layer executes the processes defined by the *DSLs* that are built in the third layer of "design".

1. In the first place, users send requests in the form of documents (publications or private messages) to the target application via the social media sites;
2. Only relevant documents mentioning the target application are taken into consideration;
3. The patterns that should be found in the documents to decide whether they are relevant or not are defined by the application designer using the first appropriate *DSL* that communicates with an external "BabalNet" dictionary;
4. Relevant documents retrieved using the first *DSL* are analyzed using different actions such as concept selection, aggregation and frequency calculation of words;
5. The actions of treatment are defined by the user of the application using the second *DSL*;
6. The execution of these actions possibly requires a communication with the Information System of the application for sending and receiving information which are used in the synthesis of the adapted answers;

[1] https://www.w3.org/2009/sparql/

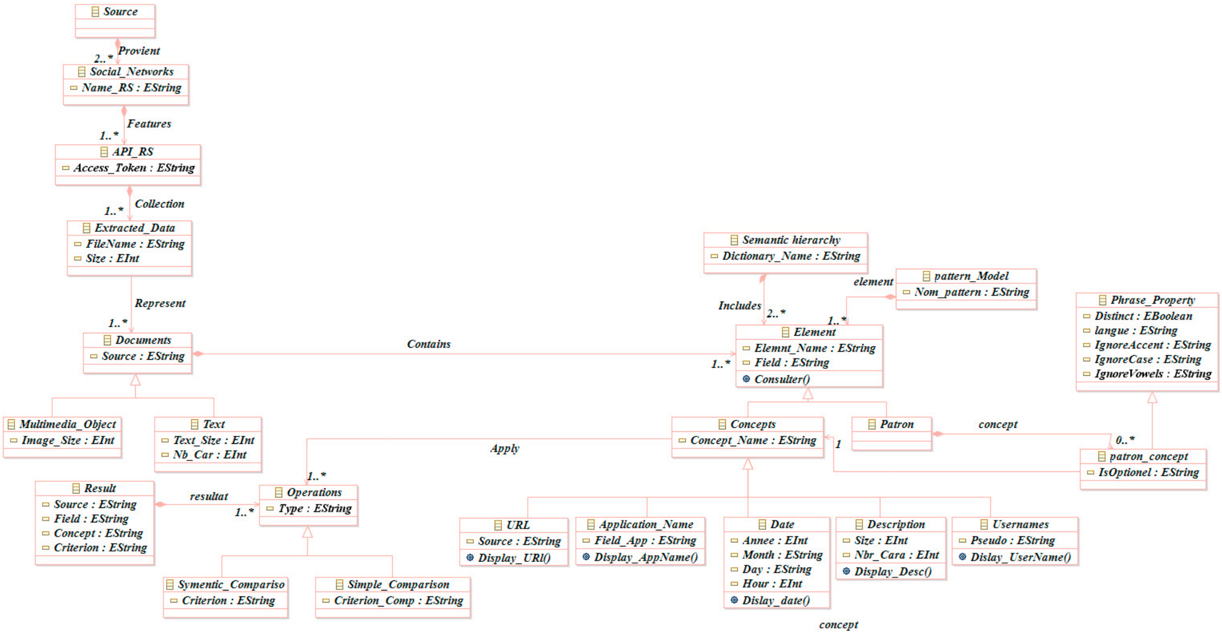


Fig. 3: An excerpt of the first *DSL* meta-model

7. The data extracted following the application of the actions or provided by the Information System are exploited in order to identify the appropriate documents to the requests of the users which are adapted to their interests. The personalization technique is defined at the second *DSL* whose purpose is to apply the personalization method based on the communities of interest;
8. Personalization uses data that are stored in the *FOAF* ontology;
9. The data extracted during the execution of the personalization are used to synthesize personalized answers;
10. Responses are sent automatically to target users as documents through social media sites.

For the development of *DSL*s, we followed the methodology proposed in [7]. It contains four phases, namely: decision, analysis, design and implementation.

3.1. *DSL* 1 : *Pattern Recognition*

In order to manipulate the documents to extract the patterns, we built a *DSL* whose meta-model is presented in *figure* 3. It is generic because it can be applied to various social media platforms. A pattern ("Pattern" class) is composed of concepts ("Concept" class), and in its simplest form a concept is a set of words. We have included specific concepts that are often used in social media like URLs, usernames and hashtags. This model is linked to an external dictionary to be able to look for the relationships of the semantic hierarchies of the concepts. Indeed, they are often linked by semantic relations (synonymy, antonymy, etc.). Patterns can be defined by the designer or extracted directly from the "BabalNet" dictionary.

3.2. *DSL* 2: *Execution of actions and personalization*

A second *DSL* for the description of actions that can be performed on documents is shown in *figure* 4. These actions are used to execute queries and select concepts from relevant documents obtained during the execution of the first *DSL*. Queries have a syntax similar to *SQL* queries. They include the selection of concepts that fulfill certain specified conditions. In addition, they make it possible to obtain certain metadata contained in the documents such as the geographical position. It also allows the communication with external information systems. This is taken into

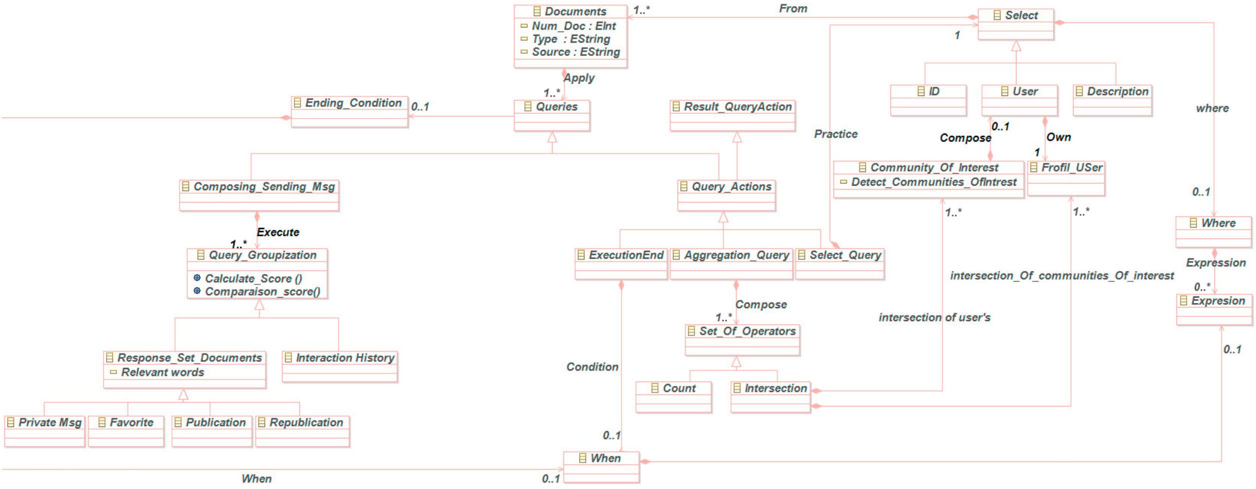


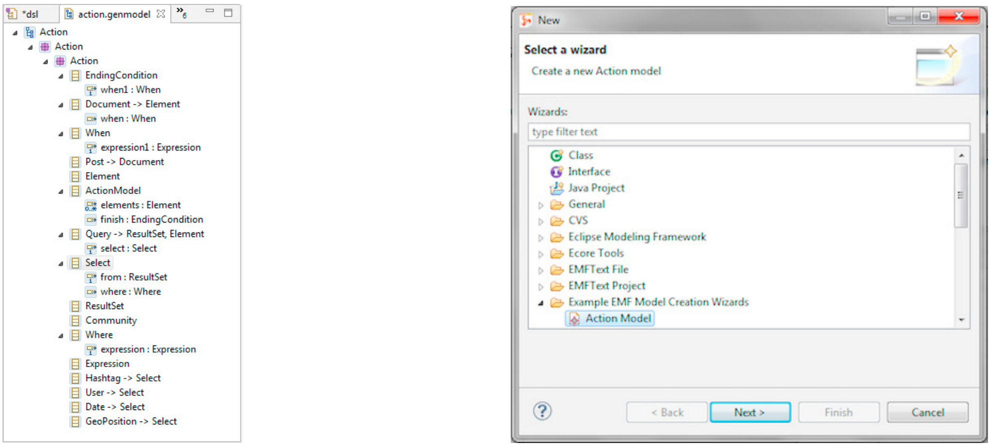
Fig. 4: An excerpt of the second *DSL* meta-model

account by allowing the definition of external data dependencies from these systems. Asynchronous events triggered by the external source can provide data to the model. Once the data are available from the queries, response documents are composed and sent to the corresponding users. This is reflected by the "Action" class. These documents may be in the form of public or private "messages". Information about the communities of interest of the users is represented by the class "CommunityOfInterest". They are extracted thanks to the communication with the ontology *FOAF*. They are used for the execution of the personalization method which is based on the "groupization" technique (class "QueryGroupization"). A response document is sent when a trigger ("When" class) is set to "true". The end of the execution of the application is signaled ("EndingCondition" class) according to a condition that may depend on several factors.

4. Implementation of *DSL* models and evaluation

The modeling steps for implementing the *DSL*s are as follows:

- Creation of the *EMF* model (.ecore extension): creation of classes, attributes and associations between classes. They exist several representations for the visualization of a model *FOAF*: view of type diagram, view of type tree or view of type text.
- Creation of the generation model (extension .genmodel): from the model defined in the previous step, it is possible to generate Java code dedicated to the creation of instances of this model. The creation of a generation model (called the genmodel) is necessary for code generation. It contains generation information such as generation path, package, and so on. ; which are not integrated into the model.
- Setting the generation model: In this step, you need to specify some parameters such as the build package ($Base_Package) and the Ecore model ($Package).
- Generating Java code and the graphical editor: Using the "Generate Model Code" utility, a set of "Java" classes are generated automatically as well as their implementations. It is also possible to make changes to the generated code such as the implementation of the body of operations and the addition of new attributes. In addition, the utility "Generate Edit and Editor Code" is used to automatically generate an editor to build model instances. Thus class instances can be built automatically via the editor.
- Instance Creation: Modelinstances can be built using directly the code generated via the editor. The modification of these instances can be done via the meta-model without code generation.



```
Action
  Action
    Action
      EndingCondition
        when1 : When
      Document -> Element
        when : When
      When
        expression1 : Expression
      Post -> Document
      Element
      ActionModel
        elements : Element
        finish : EndingCondition
      Query -> ResultSet, Element
        select : Select
      Select
        from : ResultSet
        where : Where
      ResultSet
      Community
      Where
        expression : Expression
      Expression
      Hashtag -> Select
      User -> Select
      Date -> Select
      GeoPosition -> Select
```

(a) Generation model

Select a wizard

Create a new Action model

(b) Eclipse plugin generated for instances creation

To evaluate the effectiveness of the proposed methodology, we mainly focused on evaluating the performance of the pattern recognition. The two most important measures most commonly used are: precision and recall. We applied the plugins generated following the execution of the *DSL*s by simulating an e-recruitment application. Accuracy results are reported in *table* 1.

Table 1: Accuracy of pattern recognition

	Precision	Recall
Corpus	0.96	0.98

5. Conclusion

In this paper, we proposed an *MDE* approach for automatic generation of social applications with personalized content and based on the documents generated in the social media platforms. Indeed, we defined a method of personalization of content based on communities of interest. This method is taken into account since the design phase. The proposed *MDE* architecture is based on the definition of two *DSL*s, one of which performs the personalization and the other one identifies the relevant documents. Furthermore, we plan to simulate different applications of several domains based on social media sites to ensure the personalization process.

References

[1] Bacha, Firas, Oliveira, Kathia, and Abed, Mourad. (2011) "Using context modeling and domain ontology in the design of personalized user interface." *International Journal on Computer Science and Information Systems* (6): 69-94.

[2] Kobsa, Alfred, Koenemann, Juregen, Pohl, Wolfgang. (2001) "Personalized hypermedia presentation techniques for improving online customer relationships." *The Knowledge Engineering Review* **16** (2): 111-155.

[3] Chouchani, Nadia, and Abed, Mourad. (2018) "A user-centered approach for integrating social actors into communities of interest." *12th International Conference on Research Challenges in Information Science (RCIS)* 1-11.

[4] Chouchani, Nadia, and Abed, Mourad. (2018) "Online social network analysis : detection of communities of interest." *Journal of Intelligent Information Systems* 1573–7675.

[5] Ringel, Meredith, Morris, Teevan, Jaime, and Bush, Steve. (2008) "Enhancing collaborative web search with personalization : Groupization, smart splitting, and group hithighlighting." *Association for Computing Machinery*.

[6] Teevan, Jaime, Dumais, Susan, and Horvitz, Eric. (2005) "Personalizing search via automated analysis of interests and activities." *28th Annual International ACM SI- GIR Conference on Research and Development in Information Retrieval* 449-456.

[7] Mernik, Marjan, Heering, Jan, and Sloane, Anthony. (2005) "When and how to develop domain-specific languages." *ACM Computing Surveys* **37** (4): 316-344.