A Software Factory for Accelerating the Development of Recommender Systems in Smart Tourism Mobile Applications: An Overview †

Loubna Mamad 1,*, Mamadou Mbow 2,*, Ismaïl Khriss 2 and Abdeslam Jakimi 1

1 Software Engineering and Information Systems Engineering (GL-ISI) Team, Faculty of Sciences and Techniques, University of Moulay Ismail (UMI), Errachidia 52000, Morocco; ajakimi@yahoo.fr
2 Mathématique, Informatique et Génie, Université du Québec à Rimouski (UQAR), Rimouski, QC G5L 3A1, Canada; ismail_khriss@uqar.ca
* Correspondence: l.mamad@edu.umiac.ac.ma (L.M.); mamadou.mbow@uqar.ca (M.M.)

Abstract: Recommender Systems (RSs) have become essential for suggesting personalized recommendations to users across various fields, especially in tourism. Due to the rising popularity of mobile devices, mobile RSs have emerged as a potential research area. However, developing these systems into smart tourism mobile applications requires a significant investment in artificial intelligence experts and software engineering. Hence, to reduce the cost of this investment, we propose building a software factory that provides a robust set of assets such as domain-specific languages for accelerating the development process. To this aim, we apply a model-driven engineering approach that uses models, metamodels, and model transformations to support the designing and implementation of these software systems. In this paper, we introduce an overview of our software factory to support the development of several models of RSs.

Keywords: smart tourism; recommender system; software factory; model-driven engineering; domain-specific language; model transformation

1. Introduction

To support the growth and success of the tourism domain, the tourism industry’s stakeholders use smart tourism mobile applications to improve the traveler and tourist’ experience before, during, and after their travel. Furthermore, to develop and implement these smart tourism mobile applications, several tourism destinations incorporate the 6A’s framework of Dimitrios Buhalis [1] into their applications to enhance their smart tourism destinations (STDs) management. The 6A’s framework consists of attractions, accessibility, amenities, available packages, activities, and ancillary services.

Due to several pertinent limitations including time, budgets of the tourist, and the overloading amount of available information about the points of interest (POIs), the tourist must select the suitable POIs that fit their preferences better. Hence, one of the most valuable tools and functionalities we can apply in smart tourism mobile applications to overcome these problems are recommender systems (RSs) [2]. RSs are information filtering systems that are used to provide personalized recommendations for attractions, activities, restaurants, and accommodations to users in the tourism domain based on their preferences, interests, and travel history. These systems are commonly classified into several types, including personalized RSs, content-based filtering, collaborative filtering, knowledge-based filtering, and hybrid RSs. However, developing smart tourism mobile applications that enhance the RSs for tourists is typically a complex and time-consuming process that requires a significant investment of human resources (artificial intelligence (AI) expertise,
business intelligence, and software engineering). Therefore, one of the possible solutions to overcome these issues is developing these systems through a software factory [3]. The latter aims to increase the quality, productivity, efficiency, and maintainability of the resulting software system while reducing the time and effort required for software development using a set of its assets. These assets include domain-specific languages (DSLs), frameworks, patterns, tools, and utilities for the software’s construction process analysis, design, and implementation level.

In this paper, we present an overview of our software factory to support the development of an RS.

The rest of this paper is organized as follows. In Section 2, we will present related works to our approach. In Section 3, we will introduce the overview of our proposed approach, along with a running example that illustrates the applicability of our work through a case study. Finally, Section 4 presents our conclusion.

2. Related Works

Several approaches have been proposed for automating and speeding up the development of RSs through an MDE approach in different application domains, such as modeling languages, mobile applications, business processes, and social networks.

In this regard, Almonte et al. [4] proposed an automated model-driven tool for automating the creation of RSs for low-code development platforms by using a DSL called “DROID” which configures and deploys different RSs into modeling technologies. Di Sipio et al. [5] suggested a low-code environment named LEV4REC that enables experts with no programming knowledge to design, configure, and deploy from scratch, including the parameters fine-tuning their own RSs. This tool provides a graphical interface to select RS features such as datasets, algorithms, pre-processing techniques, and evaluation metrics. Then, the final step consists of applying a model-driven code generator to create the corresponding source code of the intended RS, thereby being ready for the current deployment.

Additionally, to support the development of mobile RSs of geographic POIs from a model-driven perspective, Rojas et al. [6] introduced a conceptual framework that uses UML models. These models include UML class diagrams to express the structural specification of the RS graphically, communication diagrams to represent the behavioral specification of the recommendation algorithms, and a proprietary navigational model to describe the system’s navigational aspect. On the other hand, Rojas and Garrido [7] focused on social network-based RSs and proposed a framework for the rapid development of these systems. This paper presents several MDE concepts and techniques including the use of UML models, Model-to-Text transformation, and code generation to save time and reduce errors.

Khider et al. [8] presented a conceptual framework termed the social business process recommender (SBPR) which automatically generates business process models (recommended items) based on the defined metamodels and transformation rules. These metamodels include the user business profile metamodel and the business process metadata metamodel. All the abovementioned studies these papers used MDE as a foundation for the proposed approach.

3. Overview of the Approach

To better understand the purpose of our approach, we proposed developing a small smart tourism mobile application for the Darâa-Tafilalet region as our case study. The application in question illustrates, in particular, the functionalities related to the RSs. In this aim, our proposal was based on the use of models. In this logic, the construction of a software factory was adopted as a solution. Figure 1 provides the global overview of our software factory, with the different steps that will lead us to the mobile application’s code generation and PSM’s production. The assets that make up the foundation of our software factory are as follows:
A. The domain model (6A) based on UML, which gathers all of the different concepts of the smart tourism domain;
B. The DSL for the development of the conversational agents (CA);
C. The DSL for the development of the recommender systems (RS);
D. The DSL for the development of the user interface (UI);
E. A set of application frameworks used to implement these applications.

Each asset in the software factory is an instance of a metamodel. In this paper, we were interested in asset C, which represents the DSL to describe an RS at a higher abstraction level. The metamodel of this asset was defined using MOF meta object facility (MOF) technology [9].

Using asset C, a marketing analyst can describe the recommender functions that will be included in the mobile application. For instance, Figure 2 illustrates this description—we need recommender functions for attractions (GetAttractionsWithSameProfile), hotels (GetHotelsWithSameProfile), and events (GetSimilarPreferredEvents). The type of the two recommender functions was user based, while it was item based on the last one.
Figure 2. Using this user interface, a marketing analyst can specify the recommender functions for the mobile application. Note that the icons illustrating the different functionalities will be added later, making it easier for marketing people to understand this model.

Model C will then be transformed into a model $C'$ and merged with other models expressing other mobile application functions to obtain its PIM (platform-independent model).

Finally, the PIM will be parameterized, as shown in Figure 3. This step, performed by a developer, consists of informing the recommendation algorithms (e.g., Similarity and Nearest Neighborhood metrics) and the various design decisions related to the specification of the architecture (in our case, we use the “clean architecture”). Ultimately, our approach permits the generation of a PSM (platform-specific model) and the corresponding source code of the mobile application.
4. Conclusions

We have presented an overview of our software factory to support the development of RSs in smart tourism mobile applications. We showed several artifacts of the software factory using a case study.


Funding: This research was funded by the National Center for Scientific and Technical Research (CNRST) under a Moroccan project termed “Al-Khawarizmi program in AI and its Applications”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data have been presented in the main text.

Conflicts of Interest: The authors declare no conflict of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.