



Proceeding Paper

# A Sketch of a Model-Driven Approach to Develop Smart Tourism Mobile Applications <sup>†</sup>

Hamza Abdelmalek <sup>1,\*</sup>, Lahbib Naimi <sup>1,\*</sup> , Ismaïl Khriss <sup>2,\*</sup> and Abdeslam Jakimi <sup>1,\*</sup>

<sup>1</sup> Software Engineering & Information Systems Engineering Team, Faculty of Sciences and Techniques Errachidia, Moulay Ismail University, Meknes 50050, Morocco

<sup>2</sup> Département de Mathématiques, Informatique et Génie, Université du Québec à Rimouski, Rimouski, QC G5L 3A1, Canada

\* Correspondence: h.abdelmalek@edu.umi.ac.ma (H.A.); l.naimi@edu.umi.ac.ma (L.N.); ismail\_khriss@uqar.ca (I.K.); ajakimi@yahoo.fr (A.J.)

<sup>†</sup> Presented at the 3rd International Day on Computer Science and Applied Mathematics, Errachidia, Morocco, 13 May 2023.

**Abstract:** Several approaches have been proposed to accelerate software development. Model-driven engineering (MDE) is one approach that aims to abstract and automate the development process using models and transformations as the main artifacts. This paper presents our model-driven approach to developing smart tourism mobile applications.

**Keywords:** smart tourism; model-driven engineering; model transformation; clean architecture

## 1. Introduction

The tourism sector represents an essential source of income for a country's economy. In the case of Morocco, several investments have been made to improve the infrastructure to attract more tourists. In addition, the Moroccan Ministry of Tourism has launched an initiative to increase the visibility of Moroccan tourism destinations on social media [1]. These efforts have helped place Morocco among the most visited African countries [2]. However, much remains to be done globally, and innovative strategies are needed to increase the attractiveness of Moroccan destinations. In this sense, we have conducted a research project to create a software factory to develop mobile applications in the tourism sector. In this paper, we present our model-driven approach through a case study, where we generate the backend of smart tourism mobile applications using clean architecture.

This paper is organized as follows. Section 2 presents the background of our approach, and Section 3 describes the approach. Section 4 presents the case study, while Section 5 concludes the paper and presents future work.

## 2. Background

### 2.1. Model Driven Architecture (MDA)

MDA is an approach proposed by the Object Management Group (OMG) [3] that implements MDE and provides a set of standards for developing software using models. Its main models are the Platform Independent Model (PIM), which represents the problem domain of the system, and the Platform Specific Model (PSM), which describes the solution domain of the system. In addition, the Platform Description Model (PDM) presents the concepts of a particular platform or architecture. In the MDA process, one or multiple PDMs are applied to the PIM to create the corresponding PSMs, from which the system's source code is generated. The transition from one model to another at different levels of abstraction is performed using model transformations.



**Citation:** Abdelmalek, H.; Naimi, L.; Khriss, I.; Jakimi, A. A Sketch of a Model-Driven Approach to Develop Smart Tourism Mobile Applications. *Comput. Sci. Math. Forum* **2023**, *6*, 2. <https://doi.org/10.3390/cmsf2023006002>

Academic Editor: Mohamed Oualla

Published: 30 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 2.2. Smart Tourism

Smart tourism [4] uses technology and innovative solutions to create a personalized travel experience for tourists. In this sense, the Six As [5] framework helps developers create mobile applications that enhance the tourism experience. The six As are Attraction, Accessibility, Ancillary Services, Activities, Amenities, and Available Packages. Attraction creates personalized and compelling experiences to attract tourists, while Accessibility makes it easier to get to a destination. Ancillary Services include additional services a tourist may need, such as insurance and currency exchange. Activities provide different experiences to suit tourist interests. Amenities offer hotels, restaurants, shopping, and entertainment services. In contrast, Available Packages provide various tour packages such as adventure sports and cultural experiences. Employing the Six As in smart tourism applications can provide personalized and transparent travel experiences.

## 2.3. Clean Architecture

Clean architecture [6] is a modern approach to developing software based on domain-driven design (DDD) [7] and follows the best design principles. It allows the creation of systems independent of frameworks, user interfaces, databases, and any external third-party services. One of the main rules of the architecture is the dependency rule, which states that application dependencies must point toward the core of the application, which represents the business logic (policies), and not toward the infrastructure or UI layers (mechanisms). This dependency can be achieved through defining abstractions or interfaces in the application core and their implementations in the application infrastructure [8].

## 3. Overview of the Approach

This research project aims to develop a software factory implementing an MDE approach for smart tourism mobile applications. It consists of several assets. In this paper, we focus on the domain model and the catalog of PDM assets.

### 3.1. Domain Model

The domain model provides an overview of the components and their interactions in the smart tourism application. It uses a UML class diagram to represent the concepts and their relationships. The domain model is organized in packages according to the Six As. We find the base, attractions, amenities, available packages, user profile, ancillary services, activities, accessibility, and visit plans. In addition, it highlights the dependencies between the packages, which allows a clear understanding of the relationships and interactions between the different components and facilitates the development of robust smart tourism applications.

### 3.2. Platform Description Model

Our approach defines a PDM in two views: the UML profile [3] and a set of transformation templates. The UML profile is a mechanism that extends the UML metamodel to support a platform or domain through defining concepts and constraints between them. It is used in the PIM parameterization step to mark a model with various design decisions. Transformation templates are used in the code generation step to implement the concepts of a system architecture or platform.

To implement the clean architecture, we need to specify its PDM. For simplicity, we generate only the core layer of the architecture in the case study. The UML profile of the PDM includes concepts such as Entity, AggregateRoot, Repository, Service, and Specification. An example of a constraint is the dependency between Entity and Repository.

## 4. Case Study

To evaluate the effectiveness of the proposed approach, a case study is presented to develop the backend of a mobile application using a subset of the domain model representing the PIM of the system. In this case study, we generate the source code for the

backend of smart tourism mobile applications, using clean architecture and the Microsoft .NET 7 platform (Redmond, WA, USA). The process included architecture specification using the PDM, parameterization of the PIM, and code generation.

#### 4.1. Platform Independent Model

The PIM of the system is a subset of the domain model that manages the attractions and amenities of tourist destinations in a region. It contains nine classes and three enumerations, as shown in Figure 1. We find the *Accessibility*, *Amenity*, *AmenityRoom*, *Attraction*, *Hotel*, *Location*, *LocationImage*, *PhysicalAttraction*, and *CulturalAttraction* classes and the *PhysicalAttractionType*, *AccessibilityType*, and *RoomBedType* enumerations. The *Location* base class represents a geographic location that may interest a visitor and is accessible by foot, car, bike, and/or bus. A location can be an attraction site that tourists can visit during their trip or an amenity such as hotels where they can stay during their vacation.

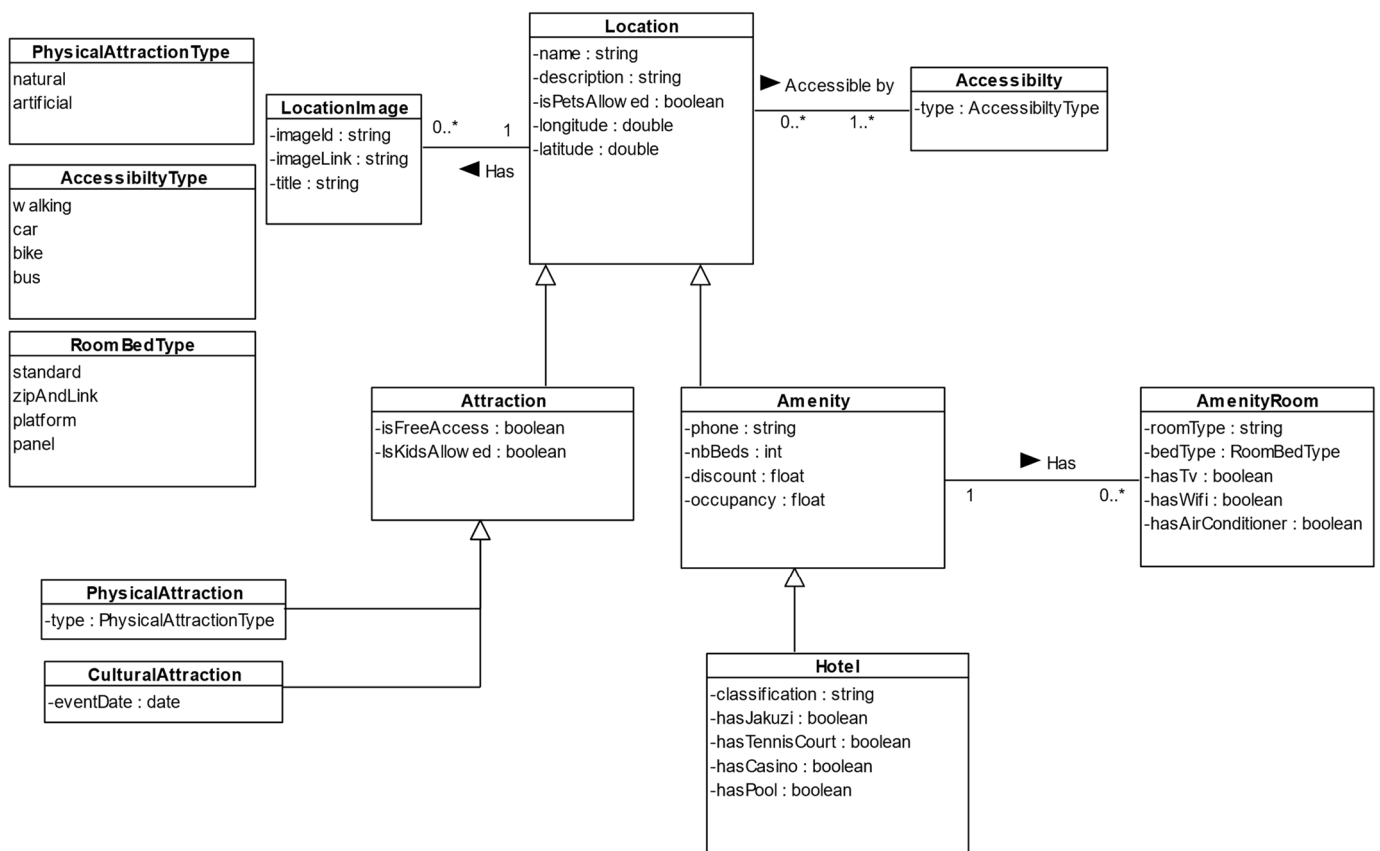
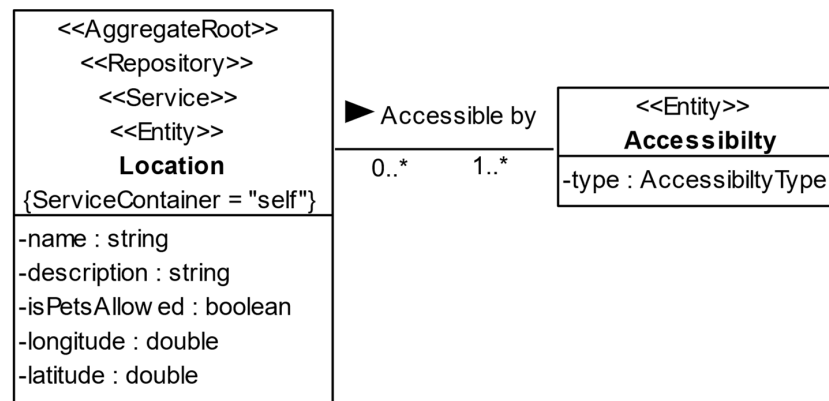


Figure 1. A PIM of the case study. UML standard: “\*” indicates the “many” multiplicity in an association. “Triangle” indicates the direction of the association.

#### 4.2. PIM Parameterization

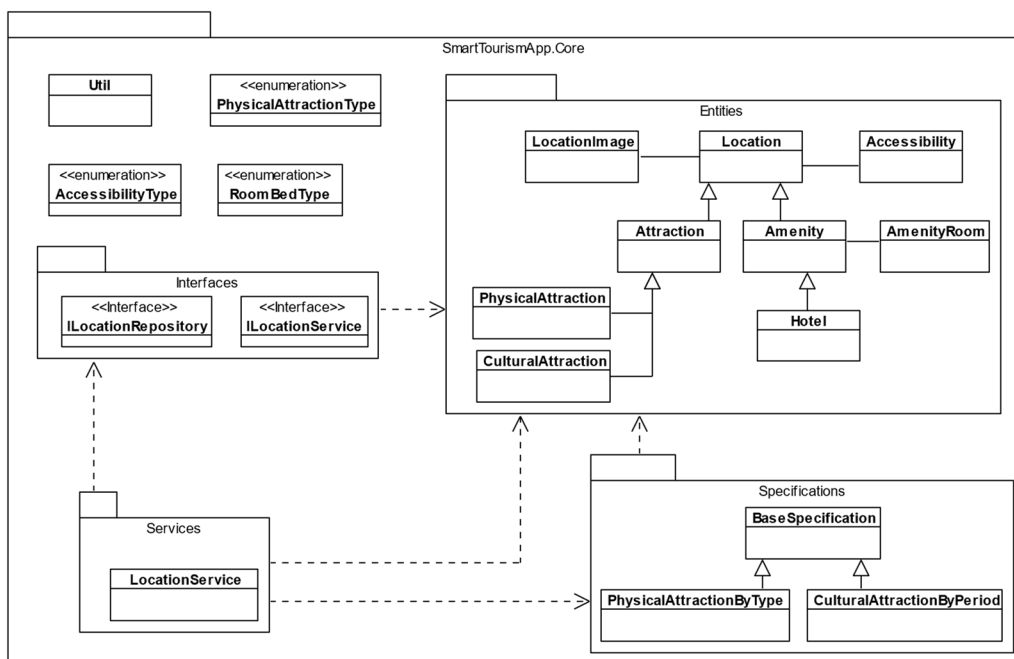
The PIM parameterization step consists of marking the PIM with design decisions from the implemented architecture. In this step, we use the UML profile of the PDM, where each defined concept is associated with a set of design concerns that specify the type of marking (stereotype or tagged value) and the concerned UML elements (class, operation, attribute, association ...). The user can simultaneously parameterize the PIM with multiple UML profiles. Figure 2 presents the parameterization of the *Location* and *Accessibility* classes with stereotypes such as *AggregateRoot* and *Entity*.



**Figure 2.** PIM parameterization of the Location and Accessibility classes. UML standard: “\*” indicates the “many” multiplicity in an association. “Triangle” indicates the di-rection of the association.

### 4.3. Code Generation

The final step in our approach is code generation, where we apply the transformation templates from the PDM to the parameterized PIM. In this step, we generate the system’s source code or the Platform Specific Model (PSM) through applying model-to-text or model-to-model transformations to the parameterized PIM, respectively. Each transformation template is applied to the model based on a condition. The condition is related to the implemented concept that specifies the concerned UML elements and the type of design concern. For example, the *Entity* transformation template, which is responsible for implementing the business logic classes of the application, is applied to the UML elements of the type *class* and parameterized with the *Entity* stereotype. Figure 3 presents the PSM of the *core* layer when the transformations of the clean architecture PDM are applied. The *core* layer is organized into four packages: the *Entities* package encapsulates the business logic of the application, the *Interfaces* package contains the definition of services and repositories, the *Services* package contains the implementation of the application services, and the *Specifications* package contains the implementations for searching entities with advanced searching criteria.



**Figure 3.** Excerpt of the core layer of the PSM.

## 5. Conclusion and Future Work

The complexity of software development can be facilitated using MDE. This paper presents a sketch of our approach to developing mobile applications to help increase Moroccan tourist destinations' attractiveness. The approach will be integrated as a software factory asset containing other artifacts needed by mobile applications such as recommender systems and conversational agents.

**Author Contributions:** Conceptualization, H.A., L.N., I.K. and A.J.; methodology, I.K. and A.J.; software, H.A. and I.K.; validation, I.K. and A.J.; formal analysis, H.A., L.N., I.K. and A.J.; investigation, H.A., L.N., I.K. and A.J.; resources, H.A., L.N., I.K. and A.J.; data curation, H.A., L.N., I.K. and A.J.; writing—original draft preparation, H.A. and L.N.; writing—review and editing, I.K.; visualization, H.A., L.N., I.K. and A.J.; supervision, I.K. and A.J.; project administration, I.K. and A.J.; funding acquisition, A.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Center for Scientific and Technical Research (CNRS) under a Moroccan project called "AI-Khawarizmi program in AI and its Applications".

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## References

1. MAP. ONMT: Le Plan D'action de Adel Fakir Pour L'exercice 2019. Available online: <https://ledesk.ma/encontinu/onmt-le-plan-daction-de-adel-fakir-pour-lexercice-2019/> (accessed on 12 March 2023).
2. Dieke, P.U. *Tourism in Africa: Issues and prospects. Sustainable Human Resource Management in Tourism: African Perspectives*; Springer: Cham, Switzerland, 2020; pp. 9–27.
3. Miller, J.; Mukerji, J. *MDA Guide Version 1.0.1*; Object Management Group, Inc.: Needham, MA, USA, 2003.
4. Gretzel, U.; Sigala, M.; Xiang, Z.; Koo, C. Smart tourism: Foundations and developments. *Electron. Mark.* **2015**, *25*, 179–188. [[CrossRef](#)]
5. Buhalis, D.; Amaranggana, A. Smart Tourism Destinations. In *Proceedings of the Information and Communication Technologies in Tourism 2014: Proceedings of the International Conference, Dublin, Ireland, 21–24 January 2014*; pp. 553–564.
6. Robert, M. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*; Pearson: London, UK, 2017.
7. Evans, E.; Evans, E.J. *Domain-Driven Design: Tackling Complexity in the Heart of Software*; Addison-Wesley Professional: Boston, MA, USA, 2004.
8. Jain, S. Clean Architecture. Available online: <https://www.linkedin.com/pulse/clean-architecture-sachendra-jain/> (accessed on 20 March 2023).

**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.