Expert Systems

WILEY

Expert Systems

REVIEW OPEN ACCESS

Enhancing Smart Tourism With Chatbots: Focus on the Metamodel of Domain-Specific Language and Emerging Technologies

Lamya Benaddi[1] | Adnane Souha[1] | Charaf Ouaddi[1] | Abdellah Chehri[2] | Abdeslam Jakimi[1] | Brahim Ouchao[1]

[1]Software Engineering and Information Systems Engineering Team, Faculty of Science and Technology Errachidia, Moulay Ismail University, Errachidia, Morocco | [2]Department of Computer Sciences and Mathematics, RMC, Kingston, Canada

Correspondence: Abdellah Chehri (chehri@rmc.ca)

Received: 2 February 2025 | **Revised:** 13 May 2025 | **Accepted:** 17 May 2025

Funding: This work was supported by Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco.

Keywords: artificial intelligence | chatbot | DSL | metamodel | smart tourism

ABSTRACT

The tourism sector is adopting smart solutions to offer visitors more personalised and sustainable experiences. By leveraging urban infrastructure and new technologies, tourist destinations aim to enhance the interaction between travellers and their environment. Artificial intelligence (AI) and natural language processing (NLP) play a key role in this transformation, particularly through chatbots. They are AI-driven applications designed to simulate human-like conversations, enabling users to interact with digital services through text or voice interfaces. In the tourism sector, they facilitate real-time access to information and services, improving the visitors' experience. These applications typically rely on intent recognition APIs, which may be proprietary, requiring access fees and potentially leading to high implementation costs. This study explores the use of a domain-specific language (DSL) dedicated to chatbot development for smart tourism. The first contribution comprises various research topics and emerging technologies used to improve smart tourism experiences and their impact on key tourism components such as attractions, accessibility, amenities, activities, available packages, and ancillary services. Second, this work aims to present the key concepts of model-driven engineering involved in constructing a DSL and to introduce our approach to building a DSL, with a focus on presenting the DSL metamodel. Third, this study identifies the challenges and limitations of using DSLs in chatbot development.

1 | Introduction

Tourism is undergoing a major transformation with the integration of emerging technologies, offering visitors more personalised, seamless, and immersive experiences while improving operational efficiency. Technologies such as artificial intelligence (AI) and natural language processing (NLP), Internet of Things (IoT), Big Data, recommender systems, augmented reality, virtual reality, and geolocation are reshaping how tourists interact with destinations and services. These innovations enable real-time access to information, dynamic service recommendations, and intelligent management of tourist flows, enhancing both visitor satisfaction and destination sustainability (Bhuiyan et al. 2022; Ouaddi, Benaddi, Jakimi, Ouchao, et al. 2024).

Despite these advancements, implementing and integrating these technologies into tourism infrastructure remains a significant challenge. Managing large volumes of data, ensuring seamless communication between different platforms, and delivering an intuitive user experience require sophisticated solutions. Among these innovations, chatbots, also called virtual assistants (Sarikaya 2017), conversational agents (Benaddi, Souha, Ouaddi, et al. 2024; Hussain et al. 2019), or bots (Lebeuf et al. 2017), have emerged as a key tool in modern tourism, offering 24/7 assistance, personalised recommendations, and automated service handling (Calvaresi et al. 2021; Benaddi, uaddi, Jakimi, et al. 2024). Chatbots are widely used for customer support, itinerary planning, booking assistance, and interactive travel experiences, providing instant responses to travellers' needs.

However, developing effective chatbots for the tourism industry is not an easy task. The process requires expertise in AI, NLP, conversation modelling, and system integration. Chatbots must handle diverse user queries, support multiple languages, integrate with third-party tourism services (e.g., hotel booking systems and transportation schedules), and deliver accurate, context-aware responses. Existing chatbot development frameworks demand significant programming skills, making it challenging for non-technical users to design and deploy chatbots tailored to tourism needs.

To address these challenges, this study explores the use of a domain-specific language (DSL) dedicated to chatbot development for smart tourism. The first contribution comprises various research topics and emerging technologies used to improve smart tourism experiences and their impact on key tourism components such as attractions, accessibility, amenities, activities, available packages, and ancillary services. Second, this work aims to present the key concepts of model-driven engineering (MDE) involved in constructing a DSL and to introduce our approach to building a DSL. This includes a detailed presentation of the DSL metamodel, structured into four packages: NLU, DM, NLG, and Tourism. Finally, the study identifies the challenges and limitations of using DSLs in chatbot development.

The rest of this article is structured as follows: Section 2 presents the theoretical background on emerging technologies, chatbots, MDE, and DSLs. Section 3 discusses related work. Section 4 presents our proposed DSL and MDE-based approach for chatbot development in smart tourism. Section 5 discusses implementation approaches for DSLs. Section 6 explores the challenges of DSL implementation in smart tourism chatbots. Finally, Section 7 concludes by summarising the contributions and discussing future research directions for improving chatbot development in the tourism sector.

2 | Fundamental Concepts and Background

This section provides an overview of the fundamental concepts and technologies that underpin this study. We examine emerging technologies in tourism, the role of chatbots in enhancing visitor experiences, and the foundations of MDE and DSLs as applied to chatbot development.

2.1 | An Overview of Chatbots

Chatbots, also known as conversational agents (Hussain et al. 2019), bots (Lebeuf et al. 2017), or virtual assistants, are software programs designed to interact with users through natural language (Adamopoulou and Moussiades 2020; Benaddi, Ouaddi, Jakimi, and Ouchao 2024a). They simulate human conversation and provide automated responses to user queries. The interest in chatbots has increased across various domains due to their potential to enhance customer service, provide information to users permanently, and improve user engagement. In the tourism industry, chatbots are particularly valuable for assisting travellers with booking services, offering travel recommendations, and providing real-time support (Benaddi, Ouaddi, Jakimi, and Ouchao 2024a).

According to the design of these agents, there are two main categories of chatbots: the first category comprises rules-based chatbots like ELIZA (Weizenbaum 1966), developed in the 1960s, which is considered the first chatbot, pioneering human-machine conversational interactions. They operate on a predefined set of rules and scripted responses and follow decision trees to provide relevant answers to user queries, making them suitable for simple interactions. The second category includes AI-based chatbots like ChatGPT. They use advanced technologies (Jbene et al. 2022a; Jbene et al. 2023; Jbene et al. 2022b; Naim et al. 2022; Raif et al. 2022; Jbene et al. 2022c; Naimi, Jakimi, et al. 2024) such as deep learning techniques, large language models (LLM), and NLP (Ouaddi et al. 2025; Souha et al. 2023) to understand and respond to more complex and unstructured inputs. These chatbots continuously learn from interactions, improving their accuracy and effectiveness over time (Figure 1).

From 2016 to 2021, chatbot development experienced significant growth due to the emergence of new platforms such as Dialogflow, Amazon Lex, and Manychat, along with frameworks like Rasa. These advancements simplified the process of building, deploying, and managing chatbots, making them more accessible across various industries. As a result, chatbots have been widely adopted to enhance customer service, automate tasks, and improve user engagement.

Figure 2 highlights the rapid increase in chatbot-related research publications, particularly after 2016, with a sharp rise in recent years. This trend reflects the growing interest in chatbot technology and its applications.

According to Figure 3, chatbot research is predominantly concentrated in computer science (33.1%), followed by engineering (13.2%), medicine (8.5%), mathematics (8.1%), and social sciences (10.2%). Additionally, chatbot applications extend to decision sciences (5.6%), business and management (4.4%), and even arts and humanities (2.5%), demonstrating their interdisciplinary impact. The increasing integration of chatbots across diverse domains highlights their adaptability and growing role in both technical and non-technical fields.

2.2 | Smart Tourism

Smart tourism enhances traditional travel experiences by incorporating advanced digital technologies such as recommender systems (Almonte et al. 2022), chatbots (Ouaddi, Benaddi, Souha, Jakimi, et al. 2024), and mobile applications

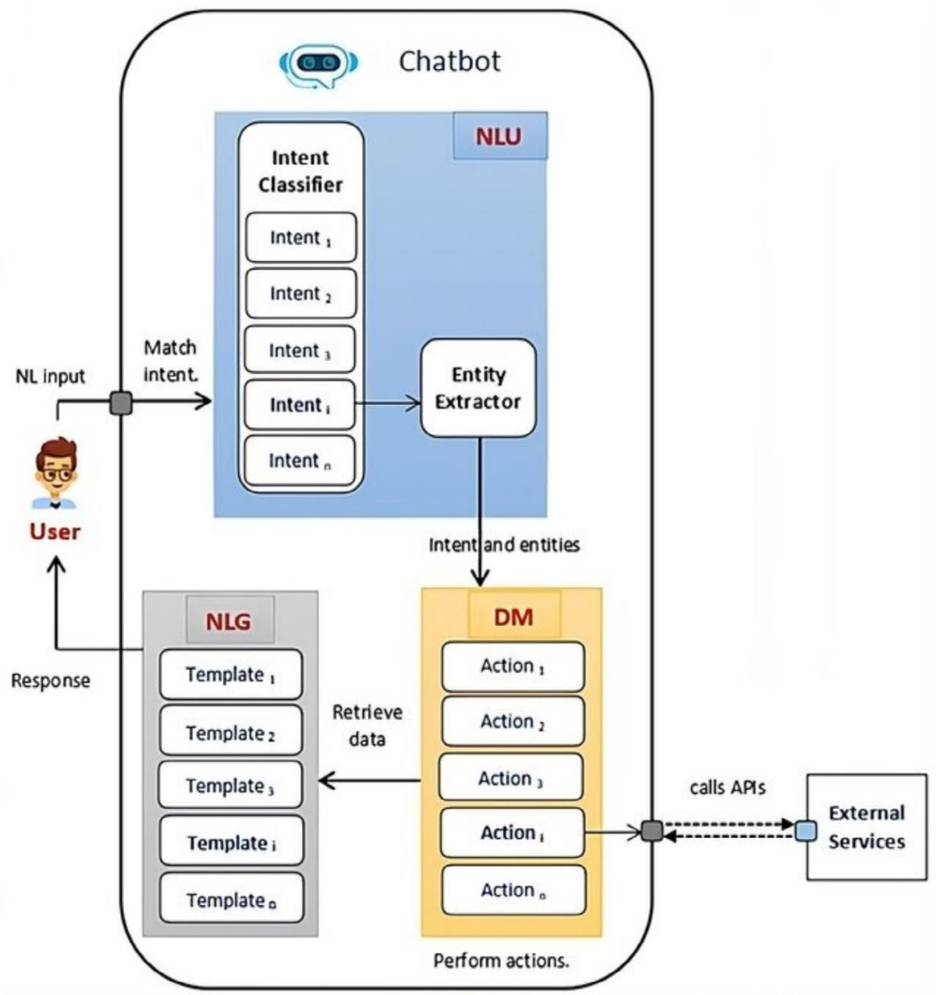

FIGURE 1 | Conceptual architecture of chatbots [adopted from (Benaddi, Ouaddi, and Souha 2024; Ouaddi et al. 2024a)].

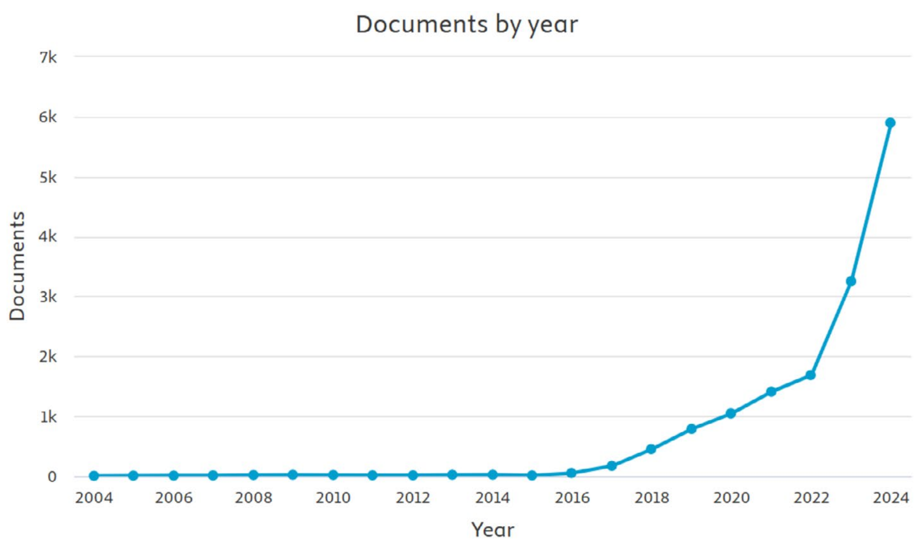

FIGURE 2 | Search results in the Scopus database from 2004 to 2024 for the keyword "chatbot*."

(Gavalas et al. 2014). These technologies enable destinations to deliver real-time information, personalised recommendations, and seamless interactions with visitors. Mobile booking platforms, interactive guides, and AI-driven suggestions help improve accessibility and engagement. Features like automatic language translation, live updates, and services for travellers with reduced mobility further enhance inclusivity. Applications like TripAdvisor provide customised itineraries and visitor reviews, ensuring a more tailored and satisfying travel experience.

The integration of these technologies has led to the emergence of smart tourism destinations (STDs), a concept that focuses on optimising tourism experiences through digital innovation (Buhalis and Amaranggana 2015). Many destinations now leverage mobile applications and AI-driven systems that align with the six core components of STDs, as outlined by Buhalis et al.: Attractions, Accessibility, Amenities, Activities, Available Packages, and Ancillary Services. These components form the foundation for designing efficient, intelligent, and engaging tourism ecosystems that improve visitor satisfaction and support sustainable tourism management.

Figure 4 illustrates the six fundamental components that contribute to a seamless and immersive travel experience. Each element plays a distinct role in enhancing tourism services through technology-driven solutions.

Attractions: This component includes tourist landmarks, historical sites, natural wonders, and cultural heritage locations.

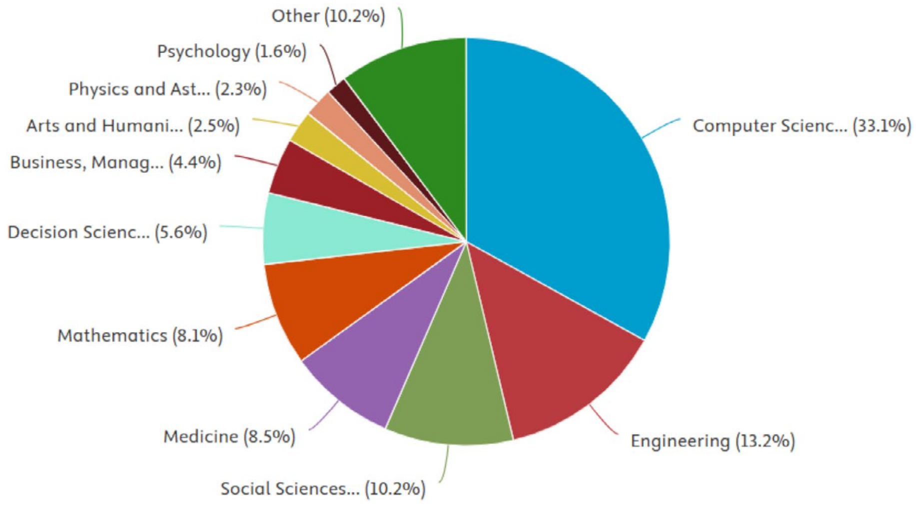

FIGURE 3 | Search results in Scopus database from 2004 to 2024 for the keyword "chatbot*" by subject area.

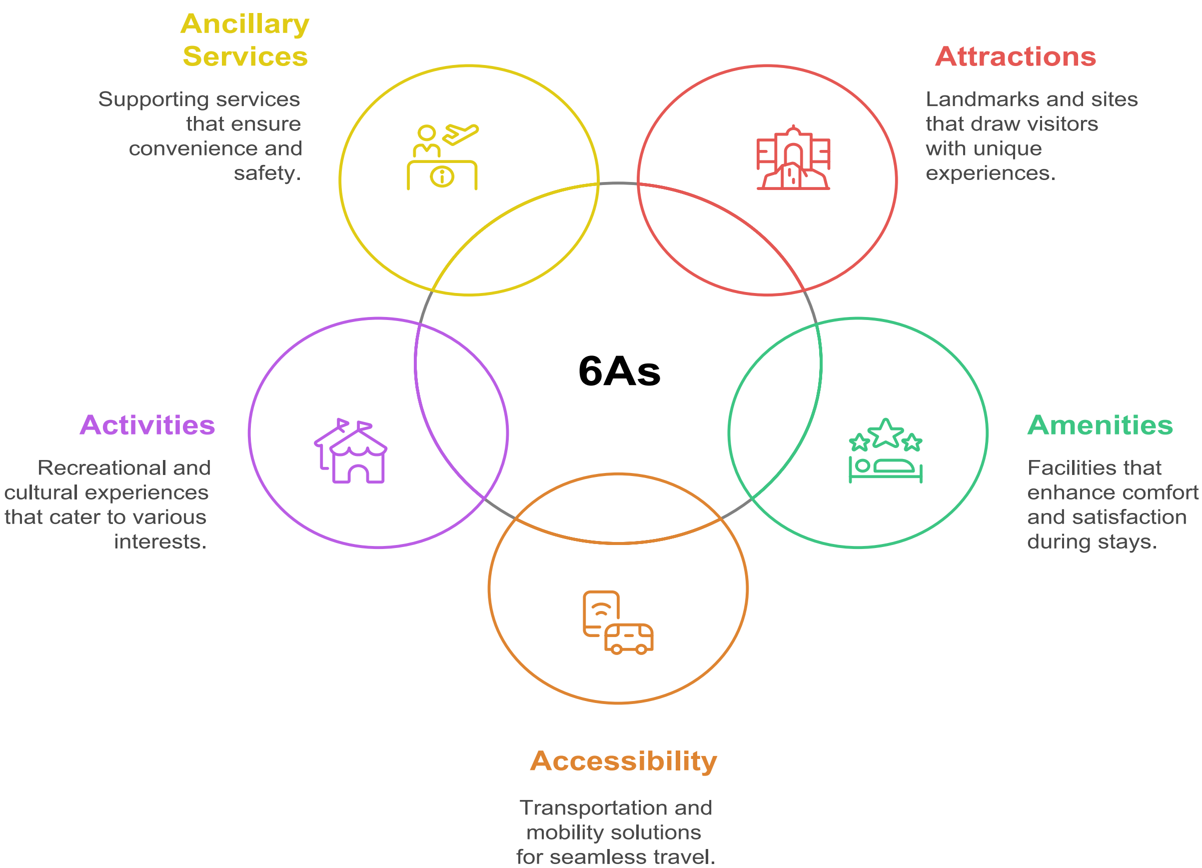

FIGURE 4 | Key components of the 6A Framework [adopted from (Benaddi, Ouaddi, Jakimi, and Ouchao 2024a)].

Attractions play a crucial role in drawing visitors to a destination, offering unique experiences that range from architectural marvels and archaeological sites to national parks and scenic landscapes. Information about attractions typically includes location details, historical significance, visitor facilities, and guided tour availability.

Amenities: These refer to the facilities and services that support tourists during their stay. They encompass accommodations (hotels, resorts, and guesthouses), dining options (restaurants, cafes, and street food vendors), and recreational spaces (spas, wellness centers, and shopping malls). Well-developed amenities enhance visitor comfort and contribute to the overall satisfaction of their trip.

Accessibility: This component focuses on transportation infrastructure and mobility solutions that enable seamless movement for travellers. It includes public transport systems (buses, metros, and trams), private transport services (car rentals and ridesharing), and accessibility features for travellers with special needs. Ensuring efficient and inclusive transport options improves connectivity between key tourism sites and enhances the overall travel experience.

Activities: A destination's appeal is often shaped by the variety of recreational, cultural, and entertainment experiences it offers. Activities can range from adventure tourism (hiking, diving, and safaris) to cultural events (festivals, exhibitions, and performances) and entertainment options (theme parks, nightlife, and live shows). Offering diverse activities ensures that tourists can engage in experiences that align with their interests and preferences.

Ancillary Services: These include supporting services that enhance convenience and safety for travellers. Examples include tour guides, translation services, travel insurance, emergency assistance, local tourism offices, and currency exchange facilities. Ancillary services provide additional layers of support, making travel more secure, accessible, and enjoyable for visitors.

Available Packages: This component covers curated travel packages, guided tours, and bundled services that help tourists plan their trips efficiently. Packages may include accommodation, transportation, sightseeing tours, and special activities tailored to different traveller preferences, such as luxury vacations, adventure tourism, eco-tourism, or cultural explorations. Well-structured packages provide convenience by offering pre-planned experiences that align with visitors' needs.

2.3 | Intelligent Chatbots for Enhancing Tourism Services

Chatbots have become an essential tool in the tourism industry, leveraging AI and NLP to enhance visitor experiences. These AI-driven virtual assistants provide instant information, personalised recommendations, and seamless customer support, making travel more efficient and accessible. As digital transformation advances, chatbots are increasingly integrated into websites, mobile applications, and social media platforms, enabling travellers to access tourism services at any time and from anywhere (Adamopoulou and Moussiades 2020).

One of the key roles of chatbots in tourism is customer service, where they handle frequently asked questions, assist with cancellations, and provide real-time support for flight or hotel inquiries. They also play a crucial role in booking assistance, helping users find and compare flights, accommodations, and attractions based on availability and personal preferences. Beyond booking, chatbots support personalised travel planning, offering tailored recommendations for itineraries, restaurants, and local experiences (Ouaddi, Benaddi, Jakimi, Ouchao, et al. 2024). Another significant advantage is their multilingual support, which helps international travellers overcome language barriers by offering translations and communication assistance in multiple languages (Jain and Bhati 2022).

Despite their growing adoption, chatbots in tourism face several challenges. One major issue is the complexity of NLP, as these virtual assistants must accurately understand user intent while managing diverse languages, dialects, and informal speech patterns. Many chatbots still struggle with context-awareness and ambiguous queries, leading to incorrect or generic responses (Ukpabi et al. 2019). Another challenge lies in integration with external platforms, as chatbots must connect with hotel booking systems, airline APIs, transportation services, and local event databases to provide accurate and up-to-date information. This requires seamless synchronisation with third-party services, which can be technically demanding and resource-intensive (Calvaresi et al. 2021).

The cost of development and maintenance is also a limiting factor. Creating AI-powered chatbots requires specialised expertise in machine learning, NLP, and software engineering, making development expensive. Furthermore, maintaining chatbots involves continuous updates to expand language models, improve conversational accuracy, and ensure compliance with evolving data security regulations (Benaddi, Ouaddi, Jakimi, and Ouchao 2024a). These financial and technical constraints make it difficult for small businesses and tourism startups to implement chatbot solutions at scale.

Despite these challenges, advancements in AI and automation are shaping the future of tourism chatbots. The introduction of more sophisticated NLP models, such as those powered by GPT-based AI, is significantly improving language comprehension and contextual awareness, making chatbot interactions more human-like. The integration of voice-enabled assistants is also gaining popularity, providing travellers with hands-free access to information through smart devices. Additionally, the combination of chatbots with IoT and geolocation technologies allows for real-time updates on local attractions, transportation schedules, and weather conditions, further enhancing the travel experience.

To address development challenges, DSLs for chatbot creation offer an innovative solution by simplifying chatbot design and reducing technical complexity. This approach enables tourism businesses to create intelligent virtual assistants without

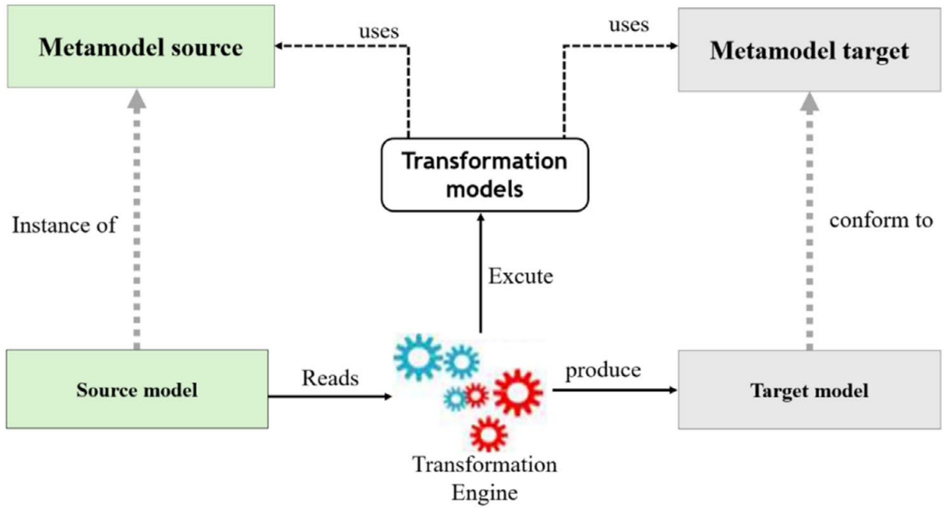

FIGURE 5 | Overview of model transformations in MDE [adopted from (Bezivin 2005; Ouaddi, Benaddi, Jakimi, Chehri, et al. 2024)].

requiring extensive programming knowledge, making chatbot technology more accessible and cost-effective (Ouaddi, Benaddi, and Jakimi 2024). As these innovations continue to evolve, chatbots will play an increasingly vital role in shaping the future of digital tourism, enhancing convenience, personalisation, and overall user satisfaction.

2.4 | MDE in Software Development

MDE is a structured approach that places models at the core of the software engineering lifecycle. Its objective is to generate system code (Bézivin and Kurtev 2005), either fully or partially, through successive transformations from models to source code. The key principles of MDE emphasise the use of models as primary artefacts to guide the software development process. These include:

- Models: Central artefacts that describe systems at an abstract level, often represented through diagrams or visual constructs, capturing the essential aspects of a system without being tied to specific implementation details.
- Metamodels: which define the structure and semantics of models.
- Model Transformations: These involve converting a model into another model or directly into source code, based on predefined templates.

The MDE principle, shown in Figure 5, revolves around the transformation between models. It is centred around the source metamodel, which establishes the conceptual framework for the source models. Based on this, the source model is generated as an instance of the source metamodel, representing the specific attributes and details of the source domain. The transformation engine then processes this model, interpreting it, applying predefined transformation templates from the transformation models, and producing the corresponding target model.

2.5 | Model-Driven Architecture (MDA)

A key implementation of MDE is MDA (OMG 2014), which has been standardised by the object management group (OMG)(Jouault et al. 2003). MDA is designed to separate business logic from technological concerns, enabling the creation of flexible and scalable systems. This separation is particularly useful for multi-platform applications, where software needs to be adapted across different environments. MDA is built on three fundamental model types:

- Platform-Independent Model (PIM): This abstract model captures the business logic and functional requirements of a system without referencing any specific technology. For instance, in chatbot development, a PIM would define the user interactions, conversation flow, and intent handling without specifying a programming language or deployment environment.
- Platform-Specific Model (PSM): Derived from the PIM, this model incorporates platform-dependent details required for implementation. In chatbot development, a PSM might include configurations for NLP frameworks such as Dialogflow or Rasa or adapt the chatbot's logic for specific platforms like Android, iOS, or cloud-based services.
- Platform Description Model (PDM): This model outlines the technical specifications and constraints of a given platform, serving as a reference for transforming PIM into PSM.

MDA also relies on automated model transformations to convert PIM into PSM and ultimately generate executable code. These transformations follow standardised rules, such as Query/View/Transformation (QVT) (OMG 2008) to describe these transformations. This architecture uses a Meta-Object Facility, known as MOF (OMG 2006), to define metamodels.

2.6 | Domain Specific Language (DSL)

A DSL is a programming language designed to address the needs of a particular domain (Cook et al. 2007). Unlike general-purpose languages such as Java or Python, which are used for a broad range of applications, DSLs provide high-level abstractions tailored to specific business or technical contexts. By aligning closely with domain requirements, DSLs simplify development, reduce complexity, and improve communication between business experts and developers. They have been widely applied in areas such as mobile application development (Lachgar and Abdali 2017), recommender systems (Souha et al. 2025), decision-making systems (Ouaddi, Benaddi, Souha, Naimi, et al. 2024; Souha et al. 2024), and web applications (Naimi, Abdelmalek, et al. 2024).

DSLs can be categorised into two main types: (i) Textual DSLs, which use scripts or structured commands to define rules, configurations, or queries. Examples include SQL for database management and languages used in automation tools; (ii) Graphical DSLs, which represent models using visual diagrams and graphical elements. Examples include UML (Unified Modelling Language) for software modelling and specialised tools for workflow automation.

- Also, A DSL as indicated in Figure 6, consists of four core components (Ouaddi, Benaddi, Bouziane, et al. 2024):
  - *Abstract Syntax*: Defines the logical structure of the DSL, focusing on its core elements rather than its visual representation.
  - *Concrete Syntax*: Specifies how the DSL is presented, whether in textual or graphical form.
  - *Semantics*: Describes the behaviour and execution rules of DSL constructs.
  - *Serializable Syntax*: Ensures that DSL constructs can be saved, exchanged, and processed consistently across different platforms.

3 | Related Works

To simplify and accelerate this process, researchers have introduced DSLs that offer specialised abstractions and enable chatbot models to be structured efficiently. For example, Conga (Pérez-Soler et al. 2021), Xatkit (Daniel et al. 2020), dFlow (Malamas et al. 2023), and the studies (Ouaddi, Benaddi, Naimi, et al. 2024; Benaddi, Ouaddi, Souha, Jakimi, and Ouchao 2024a) indicated in facilitate the design and implementation of essential chatbot components, while offering the possibility of generating code adapted to different platforms like Rasa and Dialogflow.

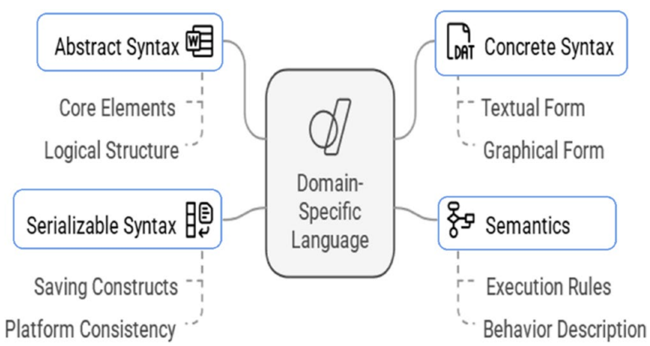

FIGURE 6 | Key components of a DSL.

In parallel, other research has focused on the definition of metamodels to structure DSLs. These works, notably those related to Bot Composer (Benaddi, Souha, Ouaddi, et al. 2024), the Rasa framework (Ouaddi et al. 2024a), and Dialogflow (Pérez-Soler et al. 2019), provide conceptual frameworks for organising the elements of conversational systems. These contributions make it possible to standardise the structure of chatbots while simplifying their customisation (Benaddi, Ouaddi, Souha, Jakimi, and Ouchao 2024b). Furthermore, the systematic literature reviews (Planas et al. 2022) (Ouaddi, Benaddi, Bouziane, et al. 2024) provide insights related to chatbot development using MDE. They have thoroughly examined existing DSLs, as well as the approaches and technologies employed in their development.

DSLs have emerged as promising tools to simplify the development of chatbots, particularly by addressing the complexities associated with NLP, conversational modelling, and integration with various platforms. Several research efforts have attempted to advance the use of DSLs in this domain. For example, Xatkit (Ouaddi, Benaddi, Naimi, et al. 2024) offers a metamodel-driven DSL focusing on three key aspects: natural language understanding (NLU) or intent metamodel, execution metamodel, and platform metamodel. This enables developers to create chatbots tailored to specific tasks.

Another significant contribution is Conga cited in (Daniel et al. 2020), which facilitates the creation of conceptual models for chatbots. It supports the generation of chatbot implementations on platforms such as Dialogflow and Rasa, bridging the gap between high-level design and practical deployment. Similarly, Ouaddi et al. (Benaddi, Ouaddi, Souha, Jakimi, and Ouchao 2024a) propose a graphical DSL that simplifies the modelling of conversational scenarios and generates the necessary project files for Rasa-based chatbots. This approach offers a visual interface for defining interactions, making it accessible to users with minimal technical expertise.

Table 1 offers a detailed comparison of various chatbot DSLs and solutions for developing chatbots, outlining the proposed solutions, the syntax used, and the implementation platforms supported for code generation.

Despite these advancements, existing DSLs for chatbot development face notable limitations. First, they do not comprehensively address the diverse range of chatbot development platforms and frameworks currently available. For instance, many frameworks, such as Amazon Lex, Microsoft Bot Framework, and others, are not fully integrated into these solutions, leaving developers to manually adapt their designs for unsupported platforms. Second, the reliance on APIs with high associated costs poses a significant barrier to adoption, particularly for small businesses and developers working within limited budgets. Moreover, existing DSLs often focus on isolated aspects of chatbot development, such as conversation modelling or NLU, without offering end-to-end support for the entire lifecycle, including deployment and maintenance. This fragmented approach limits their applicability in real-world scenarios, where interoperability and scalability are critical.

TABLE 1 | Comparison of chatbot DSLs.

Approaches	DSL solution	Abstract syntax	Concrete syntax	Platform supported by DSL
Daniel et al. (2019), Daniel et al. (2020), and Daniel and Cabot (2024)	Xatkit	Metamodels	Textual	DialogFlow
Pérez-Soler et al. (2020), Pérez-Soler et al. (2021), and Pérez Soler (2023)	Conga	Metamodel	Textual	Rasa Framework and DialogFlow
Malamas et al. (2023)	Dflow	Metamodels	Textual	Rasa framework
Ouaddi, Benaddi, Naimi, et al. (2024)	Yes	Metamodel	Graphical	Rasa framework
Benaddi, Ouaddi, Souha, Jakimi, and Ouchao (2024b), Benaddi, Ouaddi, Jakimi, and Ouchao (2024b), Benaddi, Souha, Ouaddi, et al. (2024), and Benaddi, Ouaddi, Souha, Jakimi, and Ouchao (2024b)	No	Metamodel	No	No

4 | Proposed Approach

To develop an efficient and adaptable solution for chatbot creation in smart tourism, we first conducted a comprehensive analysis of emerging technologies in the sector. This analysis aimed to identify the most impactful innovations and assess their role in enhancing visitor experiences, optimising operational efficiency, and automating tourism services. Understanding these technologies allowed us to define a DSL-based approach that leverages key advancements to improve chatbot development and integration.

In this section, we explore the emerging technologies shaping smart tourism, highlighting their contributions to personalised services, intelligent automation, and seamless user interactions. This analysis serves as a foundation for our proposed DSL-driven approach, ensuring that chatbot development aligns with industry trends and technological advancements.

4.1 | Emerging Technologies for Enhancing Tourism

The tourism industry has embraced a variety of emerging technologies to improve service efficiency and visitor engagement. Innovations such as AI, NLP, IoT, Big Data, recommender systems, augmented reality, virtual reality, and geolocation are transforming how destinations interact with travellers. These technologies enable real-time information access, personalised recommendations, and seamless digital experiences, making tourism more dynamic and visitor-centred.

However, despite their potential, the adoption of these technologies presents several challenges. The high cost of implementation makes it difficult for small businesses and developing destinations to invest in smart solutions. Additionally, technical complexity requires specialised expertise that many tourism operators lack, limiting the effective deployment of AI-driven systems, IoT infrastructure, and Big Data analytics. Data privacy and security concerns also pose risks, as collecting and analysing visitor data must comply with strict regulations.

Infrastructure limitations, especially in remote areas, can hinder the performance of digital tools, while interoperability issues create difficulties in integrating multiple platforms and services. Furthermore, resistance to change within traditional tourism businesses slows down digital transformation, and dependence on third-party providers increases operational costs and risks.

Despite these obstacles, emerging technologies provide a competitive edge for destinations that successfully integrate them. AI-powered chatbots and recommender systems enhance visitor engagement through instant assistance and personalised travel planning. IoT and Big Data optimise tourism operations by monitoring visitor flows, improving transport efficiency, and reducing congestion.

Personalised marketing strategies allow businesses to offer customised packages and promotions, increasing customer satisfaction and retention. AR and VR create immersive experiences, allowing travellers to explore destinations virtually before their visit, enhancing decision-making and engagement. Moreover, smart energy management and AI-driven analytics promote sustainable tourism by optimising resource consumption and reducing environmental impact.

These technologies have also guided the conceptual foundations of our metamodel and DSL. For example, AI and NLP inspired the inclusion of intent recognition and dialogue structure elements, while IoT and Big Data motivated the integration of contextual awareness and data-driven behaviour into the DSL's semantics. Thus, the technological landscape directly influenced the structure and expressiveness of our modelling approach.

By addressing these challenges and leveraging technological advancements, destinations can enhance their global appeal, improve visitor satisfaction, and ensure long-term sustainable growth. Table 2 summarises the key emerging technologies and their applications across various components of tourism,

TABLE 2 | Integration of emerging technologies across the 6A components of STD.

Technologies	6A					
	Attractions	**Accessibility**	**Amenities**	**Activities**	**Available packages**	**Ancillary services**
AI	Personalised recommendations for visitors	Traffic flow analysis	Optimization of hospitality services	Personalised experiences	Creation of customised packages	Optimization of ancillary services
NLP (chatbot)	Interaction via chatbots to provide information	Voice and chatbot assistance for visitors	Request management via chatbots	Chatbots for organising activities	Reservation assistance via chatbots	Real-time customer support via chatbots
IoT	Attraction flow management	Real-time transport tracking	Equipment automation (connected rooms)	Data collection for interactive experiences	Reservation monitoring	Management of local services (shuttles)
Big data	Analysis of visitor preferences	Itinerary optimization	Data analysis to anticipate needs	Activity recommendations	Creation of customised packages	Service improvement
Recommender System	Attraction suggestions based on tastes	Transportation and itinerary recommendation	Hotel and guesthouse recommendation	Activity recommendations	Personalised packages	Travel Agency recommendations, tour guide recommendation
AR (Augmented Reality)	Interactive visits	Site navigation assistance	Visualisation of amenities	Immersive experiences	—	—
VR (Virtual Reality)	Virtual visit simulations	—	—	Immersive games, educational virtual tours		—
Geolocation	Local news in real time	Optimised browsing	—	Locate nearby activities	—	—

illustrating how these innovations contribute to a more efficient and competitive industry.

4.2 | Approach Overview

Our approach introduces a conceptual framework based on the MDA paradigm to streamline and accelerate the development of intelligent chatbots for the tourism sector. This methodology emphasises flexibility, scalability, and adaptability, ensuring that chatbots cater to specific user needs and domain requirements. By leveraging the separation of business logic from technological implementation, our approach facilitates efficient chatbot creation and maintenance.

The framework follows a multi-level model-driven approach, structured around three key abstraction levels: Platform-Independent Model (PIM), Platform-Specific Model (PSM), and Platform Description Model (PDM). These models enable the progressive transformation of high-level business specifications into fully functional chatbot applications. Automated model transformations ensure a smooth and structured transition from the conceptual design phase to executable chatbot code.

At the core of this approach lies a metamodel that defines the fundamental building blocks of chatbot development. This metamodel includes essential interaction types, such as providing tourism information, making reservations, or responding to visitor inquiries. It also incorporates domain-specific entities like hotels, museums, restaurants, and local events, as well as dynamic data flows that enable chatbot responses to be tailored to user preferences, real-time location, and contextual factors.

The chatbot development process begins with the creation of PIM models, which define the chatbot's behaviour at a high level without platform constraints. These models are then refined into PSM models, incorporating platform-specific configurations such as NLP integration, chatbot deployment environments, and API connectivity. Finally, the process culminates in the generation of executable source code, ensuring that chatbots are efficiently deployed and seamlessly integrated into the smart tourism ecosystem.

Figure 7 illustrates the progression from abstract chatbot models to fully functional implementations, showcasing the structured workflow enabled by this model-driven development approach. The concretisation of our approach follows these steps:

- Designing a metamodel.
- Developing the metamodel.
- Creating a graphical interface for the DSL.
- Creating PIM models using the interface.
- Defining code generation templates.
- Performing transformations to generate the source code for various chatbot files.
- Making code adjustments and configurations.

4.3 | Metamodel of DSL

In the domain of smart tourism, the development of intelligent chatbots requires specialised tools (Ouaddi et al. 2024b) that can ensure personalisation, adaptability, and seamless interoperability with various services and platforms. To address these challenges, a dedicated DSL has been designed, enabling the platform-independent modelling of chatbots while facilitating their transformation into platform-specific implementations. This approach ensures compatibility with multiple technologies while maintaining flexibility in adapting to diverse user needs and tourism environments.

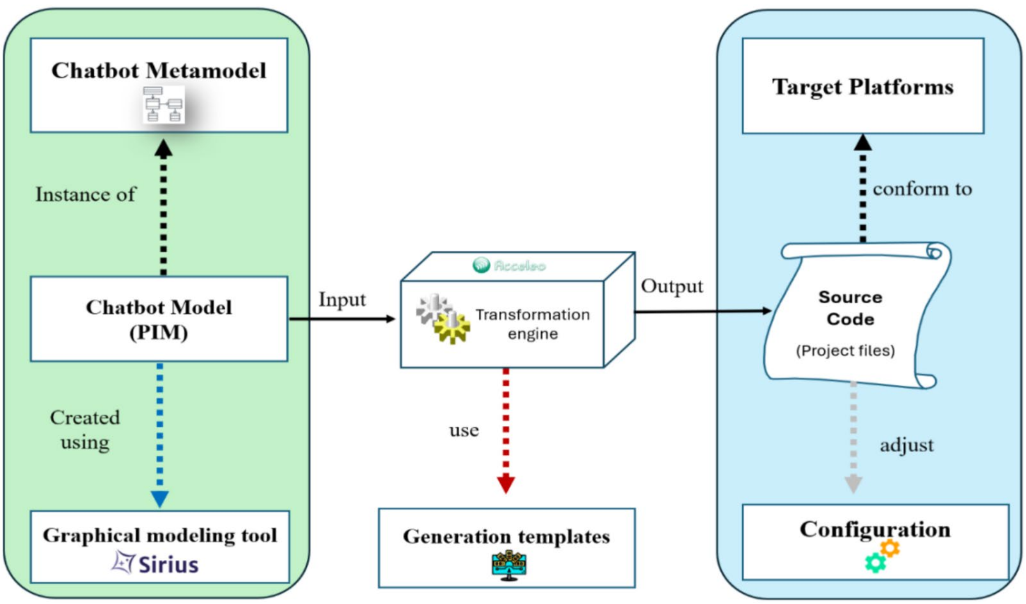

FIGURE 7 | DSL and MDE-based approach for developing chatbots.

The construction of a DSL is based on a metamodel that defines its abstract syntax. This metamodel describes chatbots at a higher level of abstraction. It is used to design chatbots providing information to users about tourist destinations in the Drâa-Tafilalet region of Morocco. A chatbot integrates the three following components: NLU, DM, and NLG. The metamodel is composed of four packages, as illustrated in Figure 8.

4.3.1 | NLU Package

The NLU package, as illustrated in Figure 9, defines the core structure for intent recognition and entity extraction within a chatbot system. The main class in this package is Chatbot, which includes attributes such as displayName to represent the chatbot's name and description to define its purpose. Additionally,

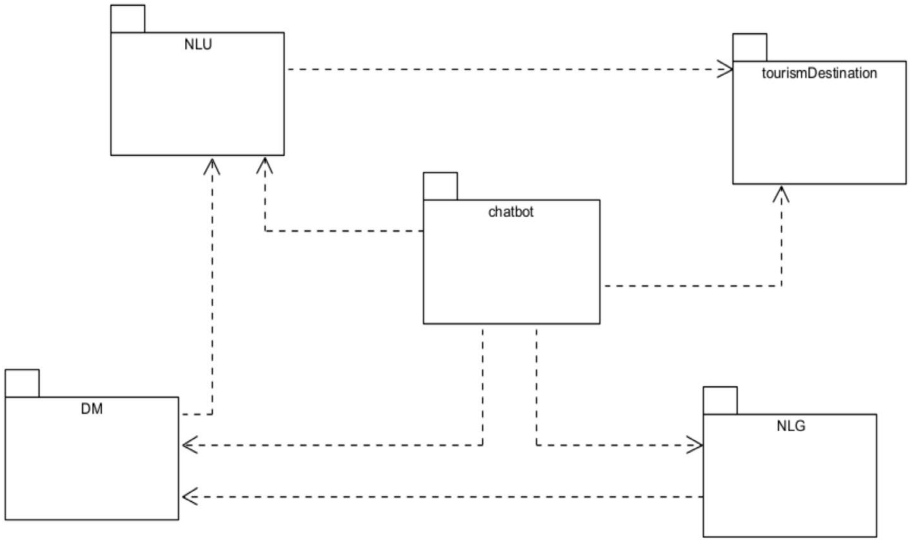

FIGURE 8 | The metamodel package.

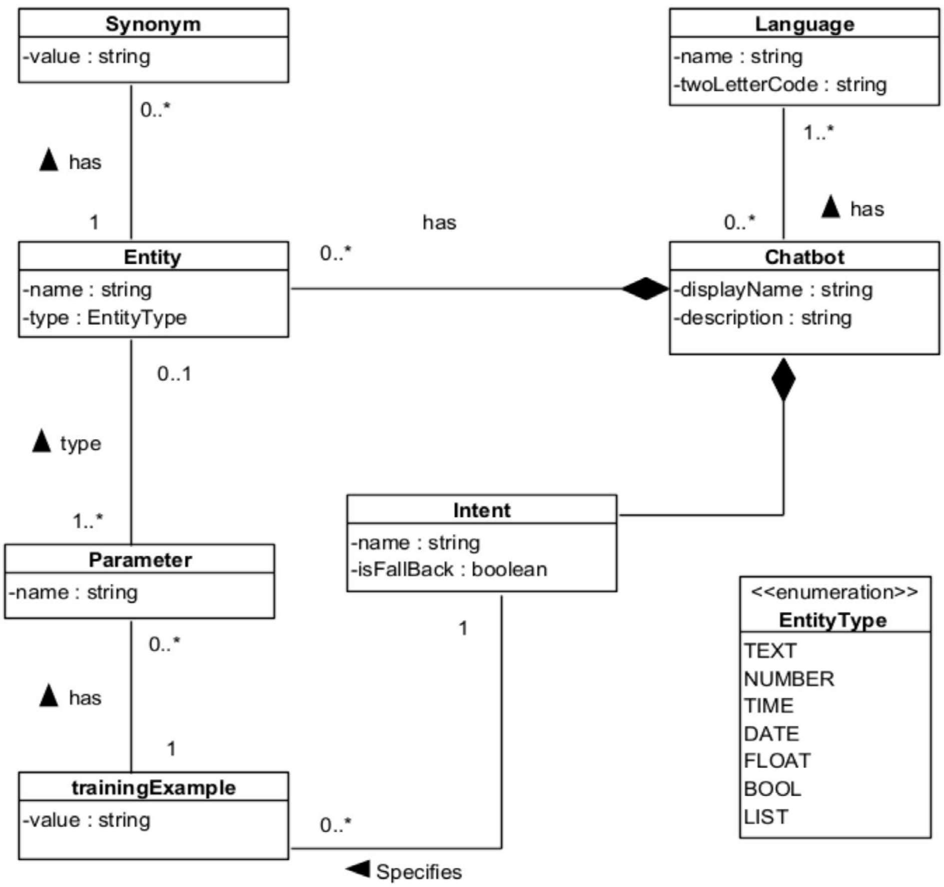

FIGURE 9 | NLU metamodel package.

a chatbot supports multiple languages, each characterised by a name and a two-letter code.

A chatbot is structured around intents and entities. The intent represents the purpose or goal behind a user's input. It allows the chatbot to understand what the user is asking for and respond accordingly. In this package, an intent has a name and an isFallBack attribute, which determines whether it serves as a fallback when no other intent matches a user's input. Furthermore, the training instances associated with an intent may contain additional information, which is stored in parameters. Each parameter has a name and an entity type. The EntityType can take several predefined values, including TEXT, NUMBER, TIME, DATE, and so on. Additionally, an Entity is a fundamental part of the intent recognition process and is characterised by its name and type. An entity helps in extracting meaningful data from user inputs and can also have multiple synonyms, allowing the chatbot to recognise different variations of a given concept or term.

4.3.2 | DM Package

The DM package, as illustrated in Figure 10, represents the Dialogue Manager (DM) of a DSL for chatbots, illustrating its key components and their relationships. The DMConfiguration serves as the main container that groups the configuration settings of the dialogue manager and manages one or more chatbots.

The DialogManager is the core system that orchestrates chatbot interactions. It contains multiple actions and conversation flows, defining the structure and progression of dialogues. Furthermore, the ConversationFlow structures the exchanges between the chatbot and the user and consists of multiple Steps. Each Step represents a turn in the conversation, incorporating both an intent and a corresponding action. Additionally, these Actions determine the responses generated by the chatbot.

4.3.3 | NLG Package

The NLG package, as presented in Figure 11, defines the structure and components required for natural language generation (NLG) within a chatbot. It ensures that responses are generated dynamically or retrieved from predefined templates using natural language. The NLGConfiguration serves as the main container that manages and configures one or more ResponseGenerators. The ResponseGenerator is responsible for producing responses by using predefined templates or dynamically generating responses based on DynamicResponse components.

Additionally, the ContextManager plays a crucial role in maintaining dialogue history, ensuring that responses remain coherent throughout the conversation. It provides contextual data to the ResponseGenerator, enhancing the accuracy of generated responses. Furthermore, the Template component stores predefined text responses, which may include placeholders for Parameters and references to Entities. This allows flexibility in adapting responses to user input. Moreover, the DynamicResponse component generates responses dynamically by leveraging external sources or real-time data, ensuring a more personalised and context-aware chatbot interaction.

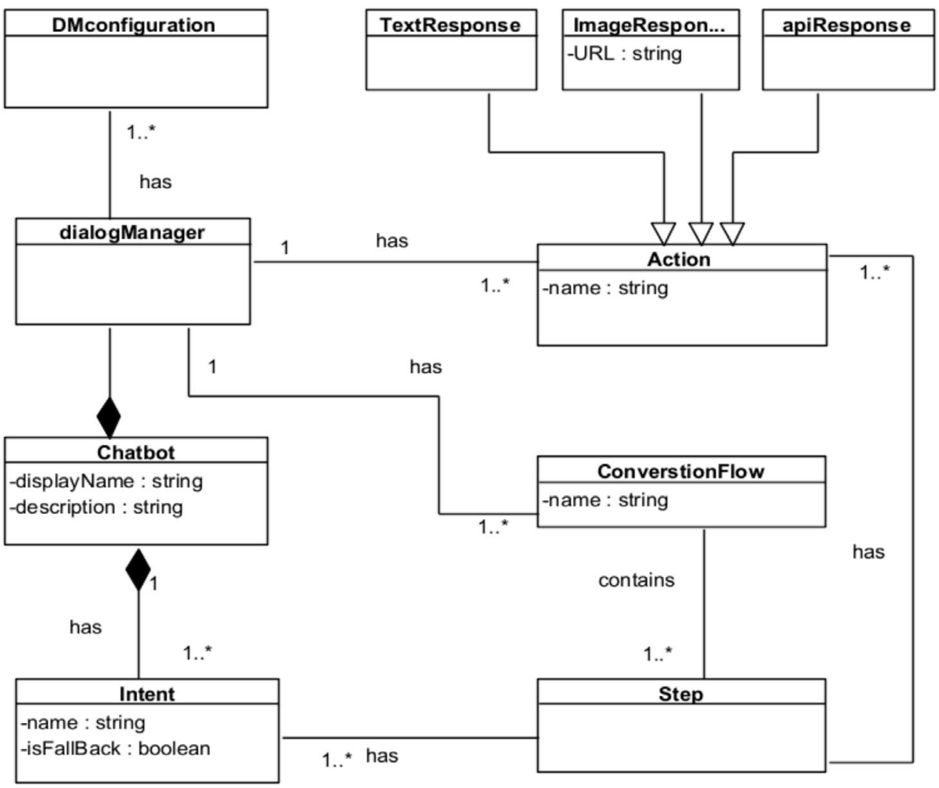

FIGURE 10 | DM metamodel package.

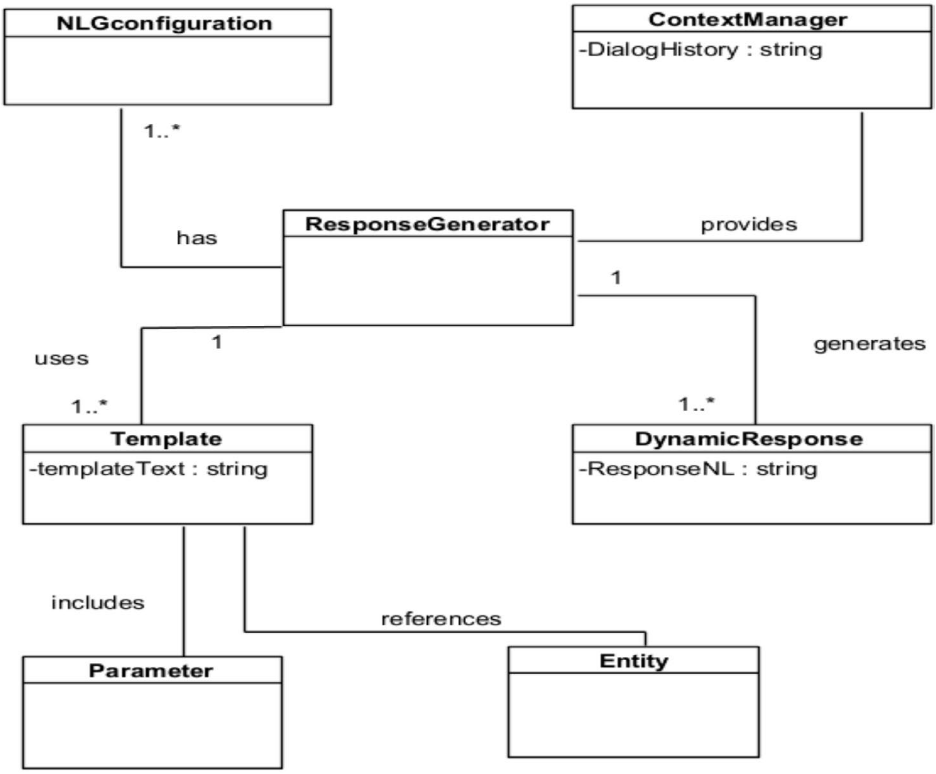

FIGURE 11 | DM metamodel package.

4.3.4 | Tourism Destination Package

The package of tourist destinations, or package 6A, represented by a UML class diagram, brings the different concepts of the smart tourism domain to provide the most complete description of the tourism sector. This package is described in Figure 12.

5 | Implementation Approaches for DSLs

The implementation of a DSL can follow different approaches, each with its own advantages and challenges. The three main methods include internal DSLs, external DSLs, and model-driven DSLs.

An internal DSL is embedded within an existing general-purpose programming language, leveraging its syntax and execution environment. This type of DSL does not require a separate parser or interpreter, as it is directly executed by the host language (Odersky et al. 2008). It is typically implemented using fluent APIs, meta-programming, or operator overloading. This approach is particularly useful for integrating domain-specific functionalities into an existing software ecosystem. For instance, Kotlin's DSL for Gradle allows developers to define build configurations concisely within the Kotlin programming language (Fowler 2010). Similarly, Python-based chatbot frameworks like Rasa utilise DSL-like constructs to define natural language processing workflows (Bocklisch et al. 2017). Despite its ease of integration and rapid development, an internal DSL is limited by the syntax and constraints of the host language, making it less flexible for complex domain-specific applications.

In contrast, an external DSL is an independent language with its own grammar, syntax, and execution model (Parr 2013). Unlike internal DSLs, external DSLs require a dedicated parser and interpreter to process commands. This approach provides greater flexibility in defining domain-specific constructs, allowing for optimised syntax tailored to the problem domain. External DSLs are often implemented using parser generators such as ANTLR or Bison, enabling the creation of a fully customised language (Chamberlin and Boyce 1974). SQL, for instance, is a widely adopted external DSL used for querying databases, offering a structured syntax specifically designed for data manipulation (Omg 2011). Similarly, Xtext provides a framework for building external DSLs with robust syntax definition and transformation capabilities. While external DSLs offer high customisation and separation from general-purpose languages, their development is more complex, requiring additional effort to design a parser, define grammar rules, and ensure integration with existing software systems.

A model-driven DSL, based on MDE, represents another approach that focuses on defining domain-specific abstractions using metamodels and transformation rules (Omg 2011). This methodology enables the automatic generation of executable code from high-level models, providing a structured and maintainable development process. The Eclipse Modelling Framework (EMF) and JetBrains MPS are commonly used for implementing model-driven DSLs, allowing users to define domain-specific structures and generate code accordingly. This approach is particularly useful in scenarios where high-level modelling is necessary, such as in business process automation or software architecture design. DSLs built on MDE principles facilitate automatic transformations, making them ideal for

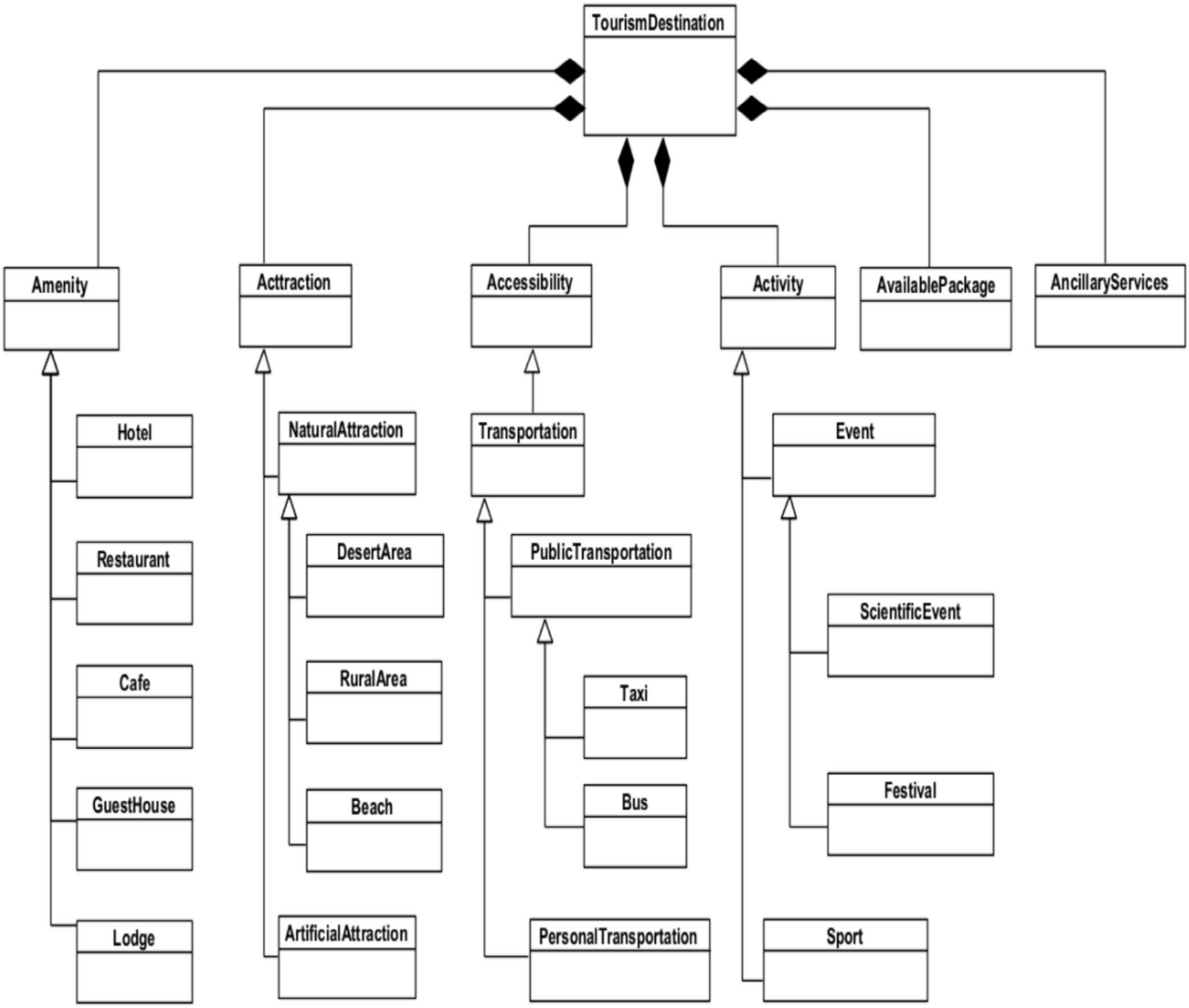

FIGURE 12 | Package of tourist destinations.

applications that require consistent and maintainable implementations (Omg 2011). However, model-driven DSLs demand expertise in modelling frameworks and often depend on specific tools, which can introduce complexity in development and deployment.

Each approach has its advantages and limitations. Internal DSLs are easy to develop and integrate but are constrained by the syntax of the host language. External DSLs provide the highest level of flexibility but require additional effort to implement parsing and execution. Model-driven DSLs enable structured design and automation but depend on specialised frameworks and tools. The choice of implementation depends on the project's requirements, the level of abstraction needed, and the complexity of the domain being addressed.

The Table 3 compares various ecosystems and frameworks for creating DSLs, including the Eclipse ecosystem, the Microsoft ecosystem, and frameworks such as ANTLR, Spoofax, JetBrains MPS, MetaEdit+, and others.

6 | Challenges of Dsl Implementation in Smart Tourism Chatbots

The implementation of DSLs for chatbot development in smart tourism presents several challenges. While DSLs offer a structured and efficient way to design and deploy chatbots, their adoption and effectiveness depend on overcoming multiple technical, linguistic, and integration-related issues. These challenges impact both the development process and the practical deployment of chatbots in real-world tourism applications.

6.1 | Complexity of NLP and Language Understanding

One of the key challenges in implementing DSLs for chatbots in tourism is ensuring accurate NLP and intent recognition. Tourists communicate in different languages, dialects, and informal speech patterns, making it difficult for chatbots to consistently understand and interpret queries. Many DSLs lack advanced NLP integration,

TABLE 3 | Comparison of ecosystems and frameworks for designing and managing DSLs.

Criteria	Eclipse ecosystem[a]	Microsoft ecosystem[b]	Other frameworks/approach
Licence	Open source	Proprietary	Open source (ANTLR, Spoofax) – Commercial (MetaEdit+)
Main objective	Creation of textual, graphical and model-based DSLs.	DSL design mainly in Visual Studio (**VS**) and related tools.	Flexibility to create external, internal or graphical DSLs.
Supported DSL types	– Textual – Graphical – Metamodel-based (EMF).	– Textual and graphical – Internal	– Textual (ANTLR, Spoofax) – Graphical (MetaEdit+) – Internal (JetBrains MPS).
Main tools	– **EMF**: Metamodel modelling – **Xtext**: textual DSLs – **Sirius**: graphical DSLs – **Acceleo**: Code generation	– **DSL Tools**: Design in VS – **Roslyn**: Syntax analysis and creation of internal DSLs – **Power Platform**: Specific workflows.	– **JetBrains MPS**: model-based DSLs – **ANTLR**: external DSLs – **Spoofax**: Rapid DSL development.
Host language/base	Java	C#, .NET	Java, Scala, Python, etc.
Community	Strong open-source community (XText, EMF, etc.).	Microsoft community focused on .NET.	Varied: depending on the tool (Spoofax, ANTLR, etc.).
Advantages	– Wide range of free tools – Flexibility for graphical and textual DSLs. – Strong support for MDE	– Integrated with VS. – Good documentation for C# developers.	– Versatility for specific cases. – Modern tools like MPS.
Disadvantages	– EMF/Sirius complexity. – Eclipse IDE is resource intensive.	– Less suitable for external DSLs. – Strong dependence on VS.	– Often requires advanced expertise. – Some tools are commercial (MetaEdit+).

[a]https://eclipse.dev/modeling/emf/.
[b]https://learn.microsoft.com/en-us/visualstudio/modeling/.

which affects chatbot efficiency, particularly in handling ambiguous questions, slang, and multi-intent sentences. Unlike traditional chatbot frameworks that rely on pre-trained machine learning models, DSL-based chatbots require manual definition of language structures, which can limit their adaptability.

6.2 | Scalability and Adaptability Across Different Tourism Services

Tourism-related chatbots must handle varied and dynamic user queries, ranging from flight bookings and hotel reservations to local attraction recommendations and transportation schedules. Developing a DSL that can flexibly accommodate multiple use cases within a single framework is a significant challenge. A rigid DSL structure may result in limited adaptability, making it difficult to expand chatbot functionalities without redesigning core components. Furthermore, tourism services frequently update pricing, schedules, and availability, requiring chatbots to be continuously synchronised with real-time databases and external APIs.

6.3 | Interoperability With Existing Platforms and APIs

Chatbots in the tourism industry often need to integrate with third-party services such as airline booking systems, hotel management platforms, tourism boards, and transport networks. A major challenge in DSL implementation is ensuring seamless interoperability with diverse APIs, databases, and software ecosystems. Many DSLs are domain-specific but struggle with cross-platform compatibility, limiting their ability to function across multiple chatbot frameworks such as Dialogflow, Rasa, and Microsoft Bot Framework. The lack of standardised communication protocols between DSL-based chatbots and external tourism services creates difficulties in data retrieval, request handling, and response generation.

6.4 | High Development and Maintenance Costs

While DSLs aim to simplify chatbot development, their initial design and implementation require significant effort. Creating domain-specific syntax, defining transformation rules, and developing execution engines demand expertise in MDE, compiler construction, and software modelling. The cost of maintaining DSL-based chatbots also increases due to the need for continuous updates, particularly in adapting to new tourist destinations, languages, and regulatory changes. Unlike general chatbot platforms, which benefit from cloud-based NLP updates, DSL-based chatbots often require manual refinement to improve understanding and response accuracy.

6.5 | Limited Adoption and Learning Curve for Developers

The adoption of DSLs in chatbot development remains limited due to their steep learning curve. Many tourism businesses rely on existing chatbot frameworks, which offer pre-configured tools and templates. Transitioning to a DSL-based approach requires training developers in custom language design, metamodeling, and model transformations. Additionally, non-technical stakeholders, such as tourism operators and business managers, may find it difficult to use DSL-based chatbot authoring tools, reducing adoption rates. The lack of user-friendly graphical interfaces further restricts accessibility for non-programmers who wish to configure chatbot interactions easily.

6.6 | Data Privacy and Security Concerns

Tourism chatbots handle sensitive user data, including personal information, payment details, and travel preferences. Ensuring secure data handling within a DSL-based chatbot framework is a major challenge. Unlike commercial chatbot platforms that integrate advanced security protocols, DSL-based solutions may lack built-in encryption, secure API authentication, and compliance with data protection regulations. The risk of data breaches and unauthorised access increases when chatbots interact with multiple third-party systems without robust security measures.

6.7 | Real-World Deployment and Performance Optimization

Once a DSL-based chatbot is implemented, real-world deployment presents additional challenges. Chatbots in the tourism sector must handle high traffic loads, particularly during peak travel seasons, requiring efficient response handling and optimised performance. Many DSLs do not provide built-in scalability mechanisms, making it difficult to maintain fast response times when dealing with large volumes of user queries. Additionally, ensuring context-aware interactions and maintaining conversation flow over multiple exchanges require advanced state management, which is often missing in early-stage DSL implementations.

7 | Conclusion

In this article, we proposed a metamodel structured into five packages (NLU, DM, NLG, and tourism destination) to construct a DSL for chatbots in the tourism sector. We also provided implementation approaches for DSLs to facilitate their practical application, and we identified the challenges and limitations of using DSLs in chatbot development. Additionally, we explored emerging technologies such as AI, NLP, IoT, and Big Data, highlighting their impact on enhancing tourism experiences and optimising visitor interactions.

Despite the benefits of using DSLs for chatbot development, several challenges remain. The complexity of NLU, the need for seamless integration with external services, and the high cost of development and maintenance are key obstacles that must be addressed. Additionally, ensuring cross-platform compatibility and optimising chatbot responses for multilingual and multimodal interactions requires further improvements.

Future research will focus on enhancing the DSL framework by integrating automated learning mechanisms to improve chatbot adaptability and context-awareness. Expanding support for multiple chatbot platforms such as Dialogflow, Rasa, and Microsoft Bot Framework will also be considered to improve flexibility. Another key direction is the development of a graphical DSL editor to simplify chatbot modelling for non-technical users. Additionally, real-world validation through case studies in STDs will be conducted to assess the effectiveness and usability of the proposed approach. Moreover, preparing a comprehensive dataset tailored to tourism domain intents is essential to provide the necessary data for training the NLU component. This dataset will serve as a foundation for improving intent recognition accuracy and enhancing chatbot performance in real-world tourism applications.

Acknowledgements

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/32).

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Adamopoulou, E., and L. Moussiades. 2020. "Chatbots: History, Technology, and Applications." *Machine Learning with Applications* 2: 100006.

Adamopoulou, E., and L. Moussiades. 2020. "An Overview of Chatbot Technology." In *Artificial Intelligence Applications and Innovations*, edited by I. Maglogiannis, L. Iliadis, and E. Pimenidis, 373–383. Springer International Publishing.

Almonte, L., E. Guerra, I. Cantador, and J. de Lara. 2022. "Building Recommender Systems for Modelling Languages With Droid."

Benaddi, L., C. Ouaddi, A. Jakimi, and B. Ouchao. 2024a. "A Systematic Review of Chatbots: Classification, Development, and Their Impact on Tourism." *IEEE Access* 12: 78799–78810. https://doi.org/10.1109/ACCESS.2024.3408108.

Benaddi, L., C. Ouaddi, A. Jakimi, and B. Ouchao. 2024b. "Towards a Software Factory for Developing the Chatbots in Smart Tourism Mobile Applications." *Procedia Computer Science* 231: 275–280.

Benaddi, L., C. Ouaddi, A. Souha, A. Jakimi, and B. Ouchao. 2024a. "Code Generation Templates for Accelerating Chatbot Development in Smart Tourism." In *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 1–5. IEEE.

Benaddi, L., C. Ouaddi, A. Souha, A. Jakimi, and B. Ouchao. 2024b. "Model-Driven Engineering to Develop Chatbots for Smart Tourism." In *2024 International Conference on Global Aeronautical Engineering and Satellite Technology (GAST)*, edited by IEEE, 1–5. IEEE.

Benaddi, L., C. Ouaddi, A. Jakimi, et al. 2024. "An Analytical Study on the Evolution and Impact of Chatbots in Tourism Over the Past Decade." In *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 1–6. IEEE.

Benaddi, L., C. Ouaddi, A. Souha, et al. 2024. "Seq2Seq Model-Based Chatbot With LSTM and Attention Mechanism for Enhanced User Interaction." arXiv preprint arXiv:2501.00049.

Benaddi, L., A. Souha, C. Ouaddi, A. Jakimi, and B. Ouchao. 2024. "Towards a Unified Metamodel for Developing the Conversational Agents for Smart Tourism." *Procedia Computer Science* 236: 241–247.

Bezivin, J. 2005. "Model-Driven Engineering: From the Principles to the Tools. Presented at the Microsoft Journées Académiques 2005."

Bézivin, J., and I. Kurtev. 2005. "MDE: Foundations, Standards and Tools. Presented at the ECDMA-FA 2005, November 2005."

Bhuiyan, K. H., I. Jahan, N. M. Zayed, et al. 2022. "Smart Tourism Ecosystem: A New Dimension Toward Sustainable Value Co-Creation." *Sustainability* 14, no. 22: 15043.

Bocklisch, T., J. Faulkner, N. Pawlowski, and A. Nichol. 2017. "Rasa: Open Source Language Understanding and Dialogue Management." arXiv preprint arXiv:1712.05181.

Buhalis, D., and A. Amaranggana. 2015. "Smart Tourism Destinations Enhancing Tourism Experience Through Personalisation of Services."

Calvaresi, D., A. Ibrahim, J.-P. Calbimonte, R. Schegg, E. Fragnière, and M. Schumacher. 2021. "The Evolution of Chatbots in Tourism: A Systematic Literature Review." In *Information and Communication Technologies in Tourism 2021: Proceedings of the ENTER 2021 eTourism Conference, January 19–22*, 3–16. Springer.

Chamberlin, D. D., and R. F. Boyce. 1974. "SEQUEL: A Structured English Query Language." In *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, 249–264. ACM Press.

Cook, S., G. Jones, S. Kent, and A. C. Wills. 2007. *Domain-Specific Development With Visual Studio Dsl Tools*. Pearson Education.

Daniel, G., and J. Cabot. 2024. "Applying Model-Driven Engineering to the Domain of Chatbots: The Xatkit Experience." *Science of Computer Programming* 232: 103032.

Daniel, G., J. Cabot, L. Deruelle, and M. Derras. 2019. "Multi-Platform Chatbot Modeling and Deployment With the Jarvis Framework." In *Advanced Information Systems Engineering*, edited by P. G. Cham and B. Weber, 177–193. Springer International Publishing.

Daniel, G., J. Cabot, L. Deruelle, and M. Derras. 2020. "Xatkit: A Multimodal Low-Code Chatbot Development Framework." *IEEE Access* 8: 15332–15346.

Fowler, M. 2010. *Domain-Specific Languages*. Pearson Education.

Gavalas, D., C. Konstantopoulos, K. Mastakas, and G. Pantziou. 2014. "Mobile Recommender Systems in Tourism." *Journal of Network and Computer Applications* 39: 319–333.

Hussain, S., O. Ameri Sianaki, and N. Ababneh. 2019. "A Survey on Conversational Agents/Chatbots Classification and Design Techniques." In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, 946–956. Springer.

Jain, A., and P. Bhati. 2022. "Comparative Analysis and Development of Voice-Based Chatbot System for Differently-Abled." *Journal of Physics: Conference Series* 2273, no. 1: 12003.

Jbene, M., S. Tigani, A. Chehri, H. Chaibi, and R. Saadane. 2023. "Tracking Dialog States in Goal-Oriented Dialogues Using a Bert-Based Siamese Network." *Procedia Computer Science* 225: 80–87.

Jbene, M., S. Tigani, A. Chehri, and R. Saadane. 2022c. "User Sentiment Analysis in Conversational Systems Based on Augmentation and Attention-Based Bilstm." *Procedia Computer Science* 207: 4106–4112.

Jbene, M., S. Tigani, R. Saadane, and A. Chehri. 2022a. "An LSTM-Based Intent Detector for Conversational Recommender Systems." In *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, 1–5. IEEE.

Jbene, M., S. Tigani, R. Saadane, and A. Chehri. 2022b. "A Robust Slot Filling Model Based on Lstm and Crf for Iot Voice Interaction." In *2022 IEEE Globecom Workshops (GC Wkshps)*, 922–926. IEEE.

Jouault, F., J. Bézivin, and M. Barbero. 2003. "J.Miller and J. Mukerji—MDA Guide Version 1.0.1 Object Management Group omg/2003-06-01. Towards an Advanced Model-Driven Engineering Toolbox."

Lachgar, M., and A. Abdali. 2017. "Modeling and Generating Native Code for Cross-Platform Mobile Applications Using DSL." *Intelligent Automation & Soft Computing* 23, no. 3: 445–458.

Lebeuf, C., M.-A. Storey, and A. Zagalsky. 2017. "Software Bots." *IEEE Software* 35, no. 1: 18–23.

Malamas, N., K. Panayiotou, and A. L. Symeonidis. 2023. "dFlow: A Domain Specific Language for the Rapid Development of Open-Source Virtual Assistants." arXiv preprint arXiv:2310.02102.

Naim, S., H. Chaibi, E. R. Abdessamad, R. Saadane, and A. Chehri. 2022. "A Hybrid Automatic Facial Expression Recognition Based on Convolutional Neuronal Networks and Support Vector Machines Techniques." In *Human Centred Intelligent Systems: Proceedings of KES-HCIS Conference*, 27–39. Springer.

Naimi, L., H. Abdelmalek, and A. Jakimi. 2024. "A DSL-Based Approach for Code Generation and Navigation Process Management in a Single Page Application." *Procedia Computer Science* 231: 299–304.

Naimi, L., A. Jakimi, R. Saadane, and A. Chehri. 2024. "Automating Software Documentation: Employing LLMs for Precise Use Case Description." *Procedia Computer Science* 246: 1346–1354.

Odersky, M., L. Spoon, and B. Venners. 2008. *Programming in Scala*. Artima Inc.

OMG. 2006. "Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 ed: OMG Document." http://www.omg.org/docs/formal/06.

Omg. 2008. "MOF 2.0 Query/View/transformation Specification. Version 1.0." www.omg.org.

OMG. 2011. "Business Process Model and Notation (BPMN), Version 2.0 ed".

OMG. 2014. "Model Driven Architecture (MDA): MDA Guide rev. 2.0." http://www.omg.org/cgi-bin/doc.

Ouaddi, C., L. Benaddi, E. M. Bouziane, A. Jakimi, A. Chehri, and R. Saadane. 2024. "DSL-Driven Approaches and Metamodels for Chatbot Development: A Systematic Literature Review." *Expert Systems* 42, no. 2: e13787. https://doi.org/10.1111/exsy.13787.

Ouaddi, C., L. Benaddi, and A. Jakimi. 2024. "Architecture, Tools, and Dsls for Developing Conversational Agents: An Overview." *Procedia Computer Science* 231: 293–298.

Ouaddi, C., L. Benaddi, A. Jakimi, A. Chehri, and R. Saadane. 2024. "A Sketch of DSL and Code Generator for Accelerating Chatbot Development." *Procedia Computer Science* 246: 3585–3594.

Ouaddi, C., L. Benaddi, A. Jakimi, B. Ouchao, and A. Abdelmounaïm. 2024. "Artificial Intelligence in Tourism Sector: Analysis Study." In *International Conference on Artificial Intelligence and Its Applications in the Age of Digital Transformation*, 61. Springer.

Ouaddi, C., L. Benaddi, L. Naimi, and A. Jakimi. 2024. "A Graphical DSL for Accelerating Chatbot Development." In *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 1–6. IEEE.

Ouaddi, C., L. Benaddi, L. Naimi, M. Rahouti, A. Jakimi, and R. Saadane. 2025. "Assessing the Effectiveness of Large Language Models for Intent Detection in Tourism Chatbots: A Comparative Analysis and Performance Evaluation." *Scientific African* 28: e02649. https://doi.org/10.1016/j.sciaf.2025.e02649.

Ouaddi, C., L. Benaddi, A. Souha, and A. Jakimi. 2024a. "Towards a Metamodel for Proactive Chatbots." In *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 1129–1133. IEEE.

Ouaddi, C., L. Benaddi, A. Souha, and A. Jakimi. 2024b. "A Comparative and Analysis Study for Recommending a Chatbot Development Tool." In *International Conference on Global Aeronautical Engineering and Satellite Technology (GAST)*, 1–6. IEEE.

Ouaddi, C., L. Benaddi, A. Souha, A. Jakimi, and B. Ouchao. 2024. "A Sketch of an Approach for Discovering UML Use-Case Diagrams From Textual Specifications of Systems Using a Chatbot." In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, 1–6. IEEE.

Ouaddi, C., L. Benaddi, A. Souha, A. Jakimi, B. Ouchao, and R. Saadane. 2024c. "Exploring and Analyzing the Impact of Chatbots in Tourism Industry." In *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, 1–6. ACM.

Ouaddi, C., L. Benaddi, A. Souha, L. Naimi, and A. Jakimi. 2024. "A Metamodel for a DSL to Design Decision Support Systems in Smart Tourism." In *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 1–5. IEEE.

Parr, T. 2013. "The Definitive ANTLR 4 Reference."

Pérez Soler, S. 2023. "Chatbots for Modelling, Modelling of Chatbots."

Pérez-Soler, S., M. González-Jiménez, E. Guerra, and J. de Lara. 2019. "Towards Conversational Syntax for Domain-Specific Languages Using Chatbots." *Journal of Object Technology* 18, no. 2: 5:1–5:21. https://doi.org/10.5381/jot.2019.18.2.a5.

Pérez-Soler, S., E. Guerra, and J. de Lara. 2020. "Model-Driven Chatbot Development." In *Conceptual Modeling*, edited by G. Cham, U. Dobbie, G. Frank, S. W. L. Kappel, and H. C. Mayr, 207–222. Springer International Publishing.

Pérez-Soler, S., E. Guerra, and J. de Lara. 2021. "Creating and Migrating Chatbots With Conga." In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 37–40. IEEE.

Planas, E., S. Martínez, M. Brambilla, and J. Cabot. 2022. "Towards Access Control Models for Conversational User Interfaces." In *International Conference on Business Process Modeling, Development and Support*, 310–317. Springer.

Raif, M., A. El Rharras, A. Chehri, and R. Saadane. 2022. "Metamorphic Testing for Edge Real-Time Face Recognition and Intrusion Detection Solution." In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, 1–5. IEEE.

Sarikaya, R. 2017. "The Technology Behind Personal Digital Assistants: An Overview of the System Architecture and Key Components." *IEEE Signal Processing Magazine* 34, no. 1: 67–81.

Souha, A., L. Benaddi, C. Ouaddi, and A. Jakimi. 2024. "A New Decision Support System for Enhancing Tourism Destination Management and Competitiveness." In *11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 1–6. IEEE.

Souha, A., C. Ouaddi, L. Benaddi, and A. Jakimi. 2023. "Pre-Trained Models for Intent Classification in Chatbot: Comparative Study and Critical Analysis." In *2023 6th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 1–6. IEEE.

Souha, A., C. Ouaddi, L. Benaddi, L. Naimi, and A. Jakimi. 2025. "MDE-Based Approach for Accelerating the Development of Recommender Systems in Smart Tourism." *IEEE Access* 13: 31615–31629. https://doi.org/10.1109/ACCESS.2025.3543058.

Ukpabi, D. C., B. Aslam, and H. Karjaluoto. 2019. "Chatbot Adoption in Tourism Services: A Conceptual Exploration." In *Robots, Artificial Intelligence, and Service Automation in Travel, Tourism and Hospitality*, 105–121. Emerald Publishing Limited.

Weizenbaum, J. 1966. "ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine." *Communications of the ACM* 9, no. 1: 36–45.