UNITECH 2023 GABROVO

INTERNATIONAL SCIENTIFIC CONFERENCE
17-18 November 2023, GABROVO

REACTIVE APPROACH FOR MULTI-AGENT TOURIST GUIDE FOR GABROVO

Iliya Iliev Nedelchev

University of Plovdiv “Paisii Hilendarski”

Abstract

This paper discusses the use of the multi-agent system Jason as a tool for implementing a tourist guide for Gabrovo. The proposed system aims to provide a personalized and human-centric experience to the tourists by considering their preferences, interests, and current location. The system uses a reactive approach that allows agents to dynamically adapt to changing situations and provide the most relevant information to tourists.

The paper argues that the reactive approach is more suitable for human-centric applications like a tourist guides, as it allows the system to respond to the tourists' needs in real-time. Moreover, the system considers the social context of the tourists by incorporating the social norms and customs of Gabrovo. The proposed system has been evaluated through a user study, which shows that the tourists find it useful and enjoyable.

In summary, this paper proposes a multi-agent system Jason for implementing a tourist guide for Gabrovo, which uses a reactive approach that is more human-centric. The system provides personalized and context-aware information to the tourists and has been evaluated positively through a user study.

Keywords: virtual tourist guide; intelligent agent for tourist guide; Jason Framework.

INTRODUCTION

Gabrovo is an idyllic city located in the heart of Bulgaria, renowned for its historical and cultural significance, as well as its captivating scenery. Nestled in the Balkan Mountains, Gabrovo offers a plethora of attractions to tourists, ranging from landmarks and museums to natural parks and outdoor activities. This dissertation delves into some of the most popular tourist destinations in and around Gabrovo, highlight-ing their unique features and cultural relevance.

One of Gabrovo's most distinguished landmarks is the House of Humour and Satire, a museum dedicated to the art of humor and satire from around the world. Po-sitioned in the city center, this museum showcases an extensive collection of cartoons, caricatures, and jokes that reflect the social, political, and cultural realities of different nations and periods. Visitors can engage in interactive exhibitions, workshops, and performances that challenge their humor and critical thinking. The House of Humour and Satire also hosts an annual international festival of humor and satire that attracts thousands of visitors from across the globe.

Another must-see attraction in Gabrovo is the Etar Open Air Museum, an excep-tional complex that recreates traditional Bulgarian life from the 19th century. Located in a picturesque valley near the village of Gabrovo, Etar features more than 50 authen-tic crafts workshops, water-powered machines, and artisan stalls that showcase the skills and tools of Bulgarian craftsmen. Visitors can explore the narrow cobblestone streets, indulge in local cuisine, and witness demonstrations of pottery, weaving, metalworking, and other traditional crafts. Etar also offers hiking trails, picnic areas, and a scenic waterfall that make it an ideal destination for families and nature enthusiasts.

For those seeking more exhilarating experiences, Gabrovo offers various outdoor activities, such as hiking, biking, fishing, and skiing.

The Central Balkan National Park, located a few kilometers from Gabrovo, features stunning natural landscapes, rare flora and fauna, and challenging hiking trails that lead to peaks, canyons, and water-falls. Visitors can also relish the breathtaking scenery of the Shipka Pass, a mountain pass that connects Gabrovo with the historical town of Shipka and provides panoram-ic views of the Balkan Mountains. During the winter season, Gabrovo transforms into a popular skiing and snowboarding destination, thanks to its modern ski resorts and well-maintained slopes.

Lastly, Gabrovo is renowned for its cultural festivals and events that reflect the city's dynamic and diverse community. The Gabrovo Carnival, held annually in May, is one of the oldest and most vibrant folk festivals in Bulgaria, featuring parades, costumes, music, and dances that celebrate the spirit of humor and satire. The National Festival of Folklore Costumes and Traditional Crafts, held in August, is another prom-inent event that showcases the beauty and diversity of Bulgarian folk traditions, with participants from all corners of the country and beyond. Other notable events in Gabrovo include the International Folklore Festival, the Jazz Fest, and the Gabrovo In-ternational Biennial of Humor and Satire in the Arts.

Gabrovo is an extraordinary and thrilling destination for those who seek to ex-plore the rich cultural and natural heritage of Bulgaria. From the House of Humour and Satire to the Etar Open Air Museum, from the Central Balkan National Park to the Gabrovo Carnival, Gabrovo offers a wealth of attractions that cater to all interests and preferences. Whether you are a history aficionado, an outdoor enthusiast, or a lover of arts and culture, Gabrovo provides a memorable and gratifying experience.

A multiagent system is a type of artificial intelligence system composed of multi-ple autonomous agents that work together to achieve a common goal. Such systems have proven to be a very good choice for implementing virtual tourist guides for cities and regions, including Gabrovo and its surrounding areas.

One of the key advantages of a multiagent system is its proactivity. The system can proactively recommend destinations and activities to tourists based on their inter-ests and preferences, making the experience more personalized and enjoyable. The agents can also work together to provide a seamless and integrated experience, providing information and recommendations in real-time as the tourist moves through different locations.

Additionally, multiagent systems are inherently more human-oriented than other types of artificial intelligence systems. This is because the agents are designed to mimic the behavior and decision-making processes of humans, including the ability to learn and adapt over time. This makes the virtual tourist guide more intuitive and responsive, which is particularly important in a tourism context where visitors may have varying levels of familiarity with the area and may require different levels of assistance.

Furthermore, a multiagent system for a virtual tourist guide can incorporate multiple sources of information, including historical, cultural, and environmental data. This allows tourists to gain a more comprehensive understanding of the area and its attractions, which can enrich their overall experience.

Certainly, a key advantage of a multiagent system for a virtual tourist guide is its ability to be responsive to changing circumstances, such as weather conditions or user behavior. For example, if the weather suddenly changes, the system can proactively adjust the recommendations and activities suggested to tourists to account for the new conditions. This could include suggesting indoor activities or attractions, recommend-ing appropriate clothing or equipment, or altering the route or schedule of the tour to avoid areas affected by the weather. Similarly, if a tourist decides to change their route or direction, the system can proactively adjust the recommendations and activities suggested to them based on their new location and preferences. The agents can work together to ensure that the tourist is still able to visit key attractions and points of in-terest, even if their original route has been altered. By being adaptable to changes in external factors and user behavior, the system can provide a more dynamic, personal-ized, and intuitive experience for tourists, enhancing their overall satisfaction and en-joyment.

EXPOSITION

Description of the system's components and their interactions:

Planner Agents: When a new tourist starts using the mobile application, a planner agent is assigned to them. The planner agent's primary task is to create a guide agent specifically tailored to the tourist's needs.

Guide Agents: The guide agents are responsible for all aspects of travel planning, including gathering information, generating plans, monitoring weather and user loca-tion, and adjusting plans as needed. They use the Jason Framework to manage their be-liefs, desires, and intentions, allowing them to adapt their plans based on various fac-tors.

Information Gathering: The guide agent collects all necessary data for travel planning using various RESTful API services. This includes data related to tourist at-tractions, and weather conditions. The agent uses this data to make informed decisions and provide relevant recommendations.

Mobile Application: The mobile application serves as the primary interface for the user to interact with the guide agent. Through the mobile app, users can input their preferences, such as budget, timeframe, and distance constraints. They can also view the generated travel plan, provide feedback, and request modifications.

Plan Generation: Based on the user's preferences and real-time data gathered from various API services, the guide agent creates an initial travel plan that considers the user's constraints and factors such as tourist attractions, accommodation, dining, and transportation.

Weather Monitoring and User Location Tracking: The guide agent continuously monitors weather conditions and the user's location using API services and GPS data from the user's device. This enables the agent to adjust the travel plan in real-time based on changing weather conditions or user movements.

Plan Adjustment: The guide agent can adjust the travel plan based on user feedback, requests, or changes in real-time data. For example, if the user decides to extend their stay in a particular location, the agent can modify the plan to accommodate this change while still considering the user's preferences and constraints.

In this system, the planner agent focuses on creating guide agents, while the guide agents handle all aspects of travel planning and communication with the user through the mobile application. The combination of Jason Framework MAS and RESTful API services enables the development of a highly adaptable and personalized tourist guide application that can respond to changing conditions and user input, ensuring an optimal travel experience tailored to each user's unique needs and preferences.

RESTful Services used and developed for the system:

- Attractions Store Service
- Tourist Service
- Map Service
- Weather Service
- Agent Environment Service

The Attractions Store Service is an advanced software solution designed to store and manage information about tourist attractions, specifically tailored for internal use by virtual tourist guides. This service is crucial in providing guides with easy access to essential data, including detailed descriptions, photo galleries, GPS locations, operat-ing hours, holiday schedules, short summaries, and other relevant metadata for each attraction.

Built on the ASP.NET 6 framework [1], the Attractions Store Service benefits from this high-performance, cross-platform web framework's features, such as rapid development cycles, extensive support for RESTful APIs, and a wide range of tools and li-braries. This choice of technology allows for the creation of a reliable, efficient, and easy-to-maintain system.

SQL Server, a scalable and secure relational database management system (RDBMS), serves as the backbone for the Attractions Store Service ref [2]. Its ability to store and manage large volumes of data, while ensuring data integrity and consistency, is vital for maintaining an extensive, up-to-date database of tourist attractions.

The Tourist Service is a specialized backend service designed to support the mo-bile application of virtual tourist guides. It plays a critical role in managing the state of tourists and establishing connections to a multi-agent system via REST. Developed us-ing the ASP.NET 6 framework, this service ensures reliability, efficiency, and seamless integration with the mobile application.

The Map Service is a specialized component designed to facilitate the interaction with Google Routes services and manage distances between various tourist attractions. This service is integral to providing accurate and efficient navigation information for users, taking into account their locations and preferred modes of transportation, such as walking or driving ref [3].

The Weather Service is another essential component of the virtual tourist guide ecosystem, responsible for obtaining accurate weather information for Gabrovo. De-veloped using the ASP.NET 6 framework, this service ensures efficiency, reliability, and seamless integration with other components in the system.

To provide users with the most up-to-date weather data, the Weather Service makes requests to WeatherAPI.com [4], a reputable and comprehensive weather data provider. By utilizing this external resource, the service guarantees access to accurate and real-time weather information, enhancing the virtual tourism experience for users who wish to consider local weather conditions when planning their virtual visits.

The Agent Environment Service is a vital component designed to facilitate communication and interaction between agents in a multi-agent system and the external environment, including the internet and the real world. Implemented using Javalin [5], a lightweight and straightforward web framework for Java and Kotlin, this service operates within the Jason multi-agent environment, providing an efficient and robust solution for agent-based systems.

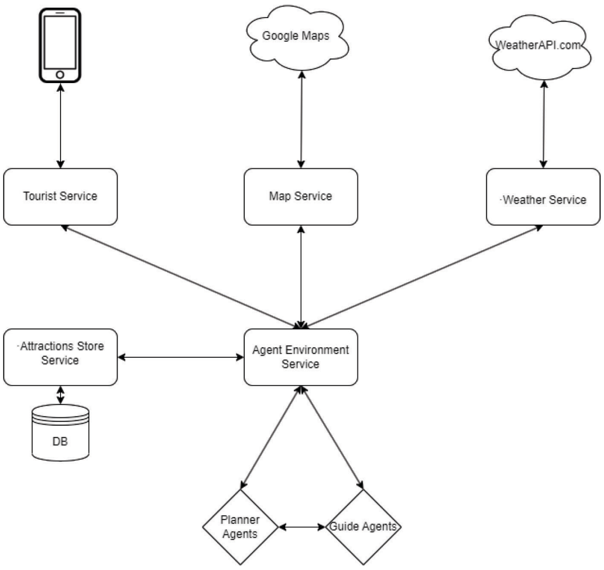

***Fig. 1**. Architecture of the services and agents*

Fig. 1 depicts the architectural framework of the relationship between various API services and multi-agent systems, highlighting how agents seamlessly interact with each other and the services to facilitate efficient coordination and task execution.

A Jason agent is an AI entity based on the Belief-Desire-Intention (BDI) model ref [6], using the AgentSpeak language. Capable of perceiving its environment, reasoning, and decision-making, the agent stores beliefs, forms desires, and commits to intentions. Jason offers a platform for creating, running, and debugging multi-agent systems, providing tools and libraries that facilitate development. Suitable for distributed prob-lem-solving, simulation, and social interaction, Jason agents enable complex, intelli-gent behavior in dynamic environments.

A planner agent is an intelligent software entity that coordinates and manages tasks by creating, monitoring, and controlling other agents or processes. Utilizing reasoning and decision-making capabilities, planner agents process incoming requests, allocate resources, and establish goals for subordinate agents. In multi-agent systems, planner agents play a crucial role in managing complex tasks and achieving system-wide objectives by dynamically adapting to changes in the environment and ad-justing the behavior of individual agents accordingly. By efficiently orchestrating tasks and resources, planner agents can enhance the overall performance and robustness of the system.

Structure of planner agent:

```
+tourist(ID,X,Y,W,B,T):true
 <- .print("new tourist");
   .create_agent(ID, "guide.asl");
   .send(ID,tell,init(ID));
   .send(ID,tell,gps(X,Y));
   .send(ID,tell,weather(W));
   .send(ID,tell,budget(B));
   .send(ID,tell,time(T));
   .send(ID,tell,startGuide(ID)).
+finishPlane(X):true
 <- .print("finish guide");
  .kill_agent(X).
```

This code snippet describes a planner agent in Jason that creates a tourist guide agent and provides it with necessary information. The code has two plans:

The first plan (+tourist(ID,X,Y,W,B,T):true) is triggered when the planner agent receives a tourist event, which contains the tourist's ID, coordinates (X, Y), weather (W), budget (B), and available time (T). Upon receiving this event, the planner agent does the following:

1. Prints "new tourist" to indicate a new request;
2. Creates a new agent with the provided ID using the "guide.asl" AgentSpeak script
3. Sends initialization data to the new guide agent, including the tourist's ID, GPS coordinates, weather, budget, and available time;
4. Instructs the guide agent to start providing guidance by sending a startGuide message with the tourist's ID;

The second plan (+finishPlane(X):true) is triggered when the planner agent receives a finishPlane event with the guide agent's ID. Upon receiving this event, the planner agent does the following:

5. Prints "finish guide" to indicate the guide's task is complete;
6. Terminates the guide agent using the .kill_agent command;

In summary, this code represents a planner agent that creates a new guide agent for each incoming tourist request, provides the guide agent with relevant information, and terminates the guide agent once its task is completed.

A guide agent is an intelligent software entity designed to assist users, such as tourists, by providing personalized recommendations and plans based on their prefer-ences, constraints, and environmental factors. Using advanced reasoning and deci-sion-making capabilities, guide agents adapt to dynamic environments and tailor their guidance to create a satisfying experience for users. In multi-agent systems, guide agents interact with other agents, such as planner agents, to effectively coordinate and execute tasks. By leveraging their knowledge base and adapting to real-time changes, guide agents deliver a high level of customization, enhancing user experiences and providing valuable assistance in various domains.

```
@init[atomic]
+init(ID): true
<- init(ID).
// New location.
+gps(X,Y): true
<- gps(X,Y).
```

```
+weather(W): true
<- weather(W).
+budget(B): true
<- budget(B).
+time(T): true
<- time(T).
+startGuide(ID): true
<- startGuide(ID);
  !createPlan.
// Finish tourist plan or time.
+finish(X): true
<- .send(planer,tell,finishPlane(X)).
+visit(A): true
<- visit(A).
+checkTime(T):true
<- adjustPlanAccordingTime(T).
+changeDirection(X,Y):true
<- adjustPlanAccordingDirection(X,Y).
+changeWeather(W):true
<- adjustPlanAccordingWeather(W).
+!createPlan:true
<- createPlan.
```

This code snippet describes a guide agent in Jason that assists tourists by creating personalized plans based on the provided information. The guide agent has several plans and rules that handle different situations:

1. Initialization (@init[atomic]): This plan initializes the agent with the tourist's ID;
2. Updating information: These plans update the agent's beliefs with new information about the tourist's location (+gps(X,Y)), weather (+weather(W)), budget (+budget(B)), and available time (+time(T));
3. Starting the guide (+startGuide(ID)): This plan is triggered when the agent re-ceives a startGuide message. It initiates the !createPlan internal action to create a personalized plan for the tourist;
4. Finishing the tour (+finish(X)): This plan is activated when the tour is complete or when the available time is over. It sends a finishPlane message to the planner agent with the guide agent's ID;
5. Visiting an attraction (+visit(A)): This plan represents the agent's belief about vis-iting a specific attraction;
6. Checking time (+checkTime(T)): This plan adjusts the tour plan based on the re-maining time;
7. Changing direction (+changeDirection(X,Y)): This plan adjusts the tour plan when the tourist changes direction;
8. Changing weather (+changeWeather(W)): This plan adjusts the tour plan based on changes in weather conditions;
9. Creating a plan (+!createPlan:true): This plan represents the internal action for creating a personalized plan for the tourist based on the available information;

The guide agent, as described in this code, collects and processes information about the tourist's preferences and constraints, and creates a tailored plan to provide a satisfying experience. It also adapts the plan dynamically in response to changes in the environment, such as time constraints, weather, or the tourist's location.

CONCLUSION

In conclusion, the Jason multi-agent system is an excellent choice for implement-ing a virtual tourist guide for Gabrovo and its surrounding region. As a well-established platform for creating, running, and debugging multi-agent systems, Jason is built upon the BDI model and the AgentSpeak language, which enable the de-velopment of complex, intelligent agents capable of reasoning and decision-making.

One of the key strengths of Jason agents is their ability to be both proactive and reactive. Proactive agents take the initiative to achieve their goals, while reactive agents quickly respond to changes in their environment. This combination allows Jason agents to adapt their behavior in real-time, ensuring a seamless and enjoyable experience for tourists. For instance, if a tourist changes direction or if weather condi-tions shift unexpectedly, the guide agent can swiftly adjust the tour plan to accommo-date these changes.

Moreover, Jason agents can easily integrate with API services, which can hold business logic and data ref [7]. This integration allows for the creation of powerful and flexible multi-agent systems that can access up-to-date information about local attrac-tions, events, and services. By combining the AI capabilities of Jason agents with the extensive data provided by APIs, the virtual tourist guide can offer personalized rec-ommendations tailored to the unique preferences and constraints of each tourist.

Additionally, the use of planner and guide agents in the Jason multi-agent system allows for an efficient and organized approach to handling tourist requests. The plan-ner agent manages incoming requests and resources, while the guide agent creates and adjusts personalized plans for each tourist. This division of responsibilities enables a more effective and scalable solution, capable of handling multiple requests simulta-neously and ensuring a high level of user satisfaction.

In summary, using the Jason multi-agent system to implement a virtual tourist guide for Gabrovo and the surrounding region provides a robust, adaptable, and intelligent solution that can enhance the overall experience for visitors. The proactive and reactive nature of Jason agents, coupled with their ability to integrate with API services, allows for the creation of personalized, real-time recommendations that cater to each tourist's needs. Furthermore, the efficient coordination of planner and guide agents ensures a well-organized and scalable system, making the Jason multi-agent system an ideal choice for creating a virtual tourist guide in Gabrovo and beyond.

REFERENCE

[1] What’s new in ASP.NET Core 6.0. Available online: https://learn.microsoft.com/en-us/aspnet/core/release-notes/aspnetcore-6.0 (23 September 2023)

[2] What is SQL Server. Available online: https://www.sqlservertutorial.net/getting-started/what-is-sql-server/ (23 September 2023)

[3] Routes API | Google Developers. Available online: https://developers.google.com/maps/documentation/routes/overview (23 September 2023)

[4] Weather and Geolocation API. Available online: https://www.weatherapi.com/docs/ (23 September 2023)

[5] Creating REST Microservices with Javalin. Online: https://www.baeldung.com/javalin-rest-microservices (23 September 2023)

[6] Bordini, R.; Hubner, J.; Wooldridge, M., Programming multi-agent systems in AgentSpeak using Jason; Publisher: Wiley 2007; pp. 15–16.

[7] Stoyanov, S.; Glushkova, T.; Doychev, E., Cyber-Physical-Social Systems and Applications: Part 1: Reference architecture; Publish-er: LAP LAMBERT Academic Publishing (December 5, 2019); pp. 5–6.