Multi-Agent Coordination Project Report

Group Members:

Jadrina De Andrade E Silva

Tareq Md Rabiul Hossain CHY

Nushrat JAHAN

Group Number: 1

Submission Date:

25/01/2024

Course Title:

Multi Agent Coordination

Instructor:

Flavien Balbo

Olivier Boissier

Luis Gustavo Nardin

1 Introduction

This report details our work done for the Multi-Agent Coordination Project. Set against the backdrop of the tourism industry's evolving needs, our project utilizes a network of virtual agents to streamline the coordination and execution of holiday packages. These agents, representing tour operators, consumers, and service companies, collaborate in a multi-stage process to enhance the efficiency and effectiveness of holiday planning.

In the following sections, we delve into the specifics of our project. The 'Code and Development' section describes the technical aspects and the strategic implementation of our system across three key phases: Contracting, Purchasing, and Delivering. Each phase is discussed in detail, highlighting the interactions among different agents and their roles in the system.

The subsections under 'Code and Development' provide a comprehensive breakdown of each phase. The Contracting phase outlines the initial negotiations and agreements among agents, the Purchasing phase focuses on how holiday packages are bought by consumers, and the Delivering phase covers the execution and delivery of services. Additional subsections detail the organizational aspects and the specific setup of our simulations.

Finally, in the 'Final Remarks' section, we reflect on the potential enhancements and future directions of our project. This includes ideas for advancing our auction mechanism and further fine-tuning the system for more complex scenarios.

Our goal with this report is not only to document our project but also to offer insights and methodologies that could be beneficial in the broader context of multi-agent systems and the tourism industry.

2 Code and development

2.1 Phase 1: Contracting phase

For this phase we started off with one TOA and one SCA and built a system for them to exchange messages. Then we added more SCAs and finally another TOA to the system. Since we had previously worked with an auction mechanism in the house building example, we decided to reuse it for this project. This phase is executed in the following steps.

1. The service company agents contact the tour operator agents with their provided services and their location.

2. The TOAs then contact only the SCAs that have the services they need and the correct location. Upon contact they let them know which of their services is needed.

3. Based on the contact the SCA can find the artifact(AucServ) and places the bid.

4. Finally after waiting for all bids to be placed the TOAs then let the winners of their auction know that they won.

5. Since the SCA could have won multiple auctions they will accept if that is the highest profit they could have and decline if they have a better offer.

6. If the SCA declines the TOA contacts the runner-up.

If the runner-up declines then the execution will stop there.

In order for our simulation to be more realistic, each SCA is generated in such a way that the services they offer, their prices for these services, and their location are random (sca.asl). We create two of these agents. And to make sure that there is always at least one service company offering the service we made a second file (sca_f.asl) which is a SCA with the three different types of services at fixed prices and a fixed reputation. We create three instances of this type of agent to have one for each possible location. For the TOA we made two instances using the same file (toa.asl) and the only thing that is randomized is the location.

2.2 Phase 2: Purchasing phase

For the second phase we use a very similar approach to the first phase. We also use an auction mechanism though this time the bidders are the TOAs and the one offering is the client. This phase is executed in the following steps.

1. The tour operator agents send their location to the two clients.

2. The client then contact only the TOAs that have the correct location.

3. Based on the contact the TOA can find the artifact and places the bid.

4. Finally after waiting for all bids to be placed the clients then let the winners of their auction know that they won.

5. The TOA then lets the service companies they hired know who the client is.

We use one file to create the two clients (ca.asl), bob and bobbette, who each have different locations. Each client creates their own PrefServ artifact using their name so that the TOAs can easily focus on it and place their bids. Since the two clients are in different locations, we do not have to worry about a TOA winning two auctions. If no TOAs are available in the client's location the message ”No TOA meets my criteria” appears on the console and the execution is forced to fail.

2.3 Organization

In our project, we've focused on creating holiday trip plans that cater to different traveler preferences. We conceptualized three holiday packages: Basic, Medium, and Premium. However, it's important to note that as of now, the Premium package is a proposed concept and has not been fully implemented in our system. The Basic and Medium packages are operational and interact effectively with the agent we have developed.

Figure 3 below shows the structural specification of our organization, providing a visual overview.

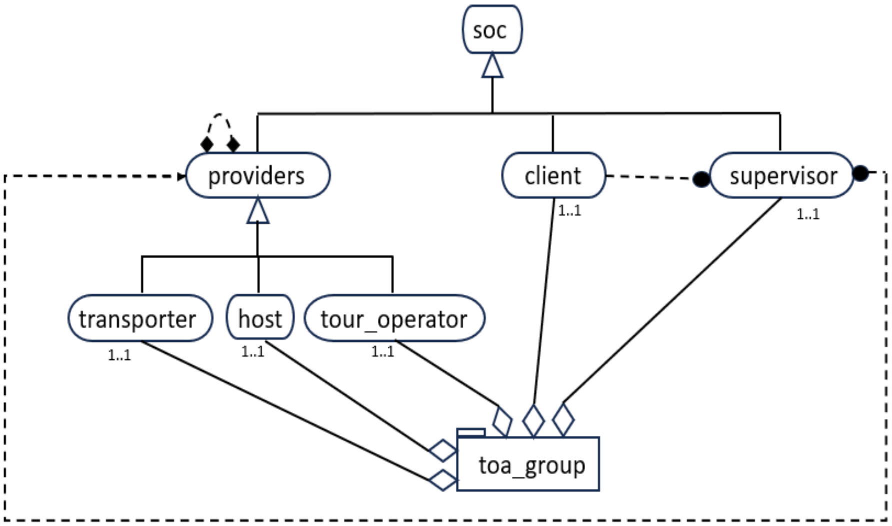


Figure 1: Structural Specification of Organization

2.3.1 Basic package

The Basic package offers the essentials of travel, focusing on simplicity and convenience. It includes:

1. Airport to accommodation transport: A reliable and comfortable transfer service from the airport to the accommodation.
2. Accommodation service: Quality lodging options ensuring comfort and safety for travelers.
3. Return transport to the airport: Ensuring timely and stress-free transportation back to the airport at the end of the stay.

This package is ideal for travelers who prioritize efficient and straightforward travel arrangements.

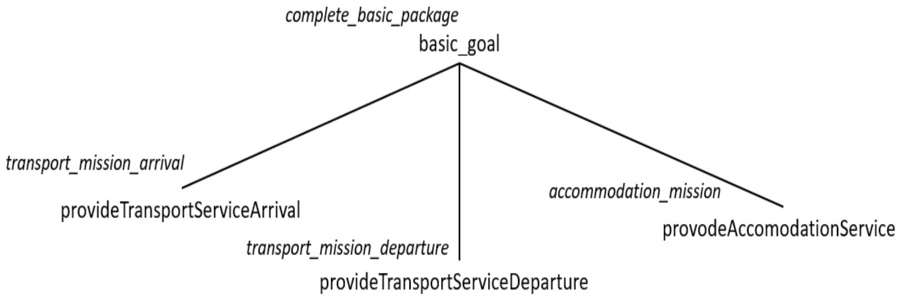

complete_basic_package
basic_goal
transport_mission_arrival
provideTransportServiceArrival
transport_mission_departure
provideTransportServiceDeparture
accommodation_mission
provodeAccomodationService

Figure 2: Functional Specification of Basic Organization

2.3.2 Medium package

Building on the Basic package, the Medium package adds an element of exploration and includes:

1. Accommodation service: Similar to the Basic package, offering comfortable and secure lodging.

2. Transport to tour location: Convenient transportation services from the accommodation to a pre-selected tour location, enhancing the travel experience.

3. Tour service: An engaging and informative tour, offering insights into local culture, history, or nature.

4. Return transport to accommodation: Seamless and comfortable transportation back to the accommodation post-tour.

This package is designed for travelers seeking a blend of comfort and adventure in their journey.

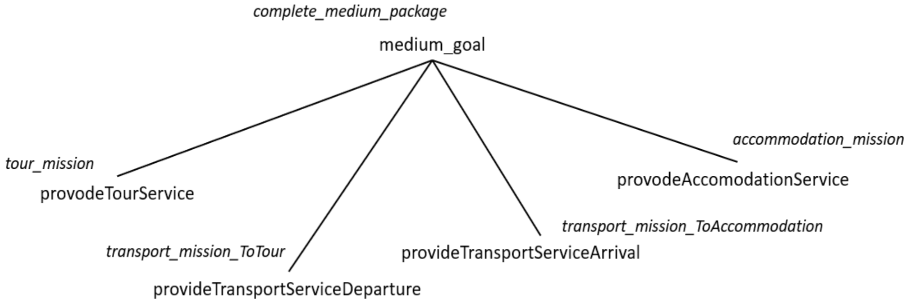

complete_medium_package
medium_goal
tour_mission
provodeTourService
transport_mission_ToTour
provideTransportServiceDeparture
provideTransportServiceArrival
transport_mission_ToAccommodation
accommodation_mission
provodeAccomodationService

Figure 3: Functional Specification of Medium Organization

2.4 Phase 3: Delivering phase

The third phase is triggered by the organization, and the goals are carried out by the service company agents that provide the service and the clients that print messages to let the user know what is happening. In the plan for go, the TOA will create the organization that corresponds to their offered package and then it will send a message to the service companies and their client in order for them to adopt their respective roles in the package. After that we have several goals that correspond the missions assigned to the goal. These goals which all have the from provideXService, allow the SCA to create a ServPlace artifact, that guarantees that only one service company can carry out that service in that location at the time. Before creating the artifact, the agent will check if it already exists, if it does then it skips the creation step. Then the SCA sends a message to the client so they can focus on that artifact. Finally, they send another message to let them know they can use the service.

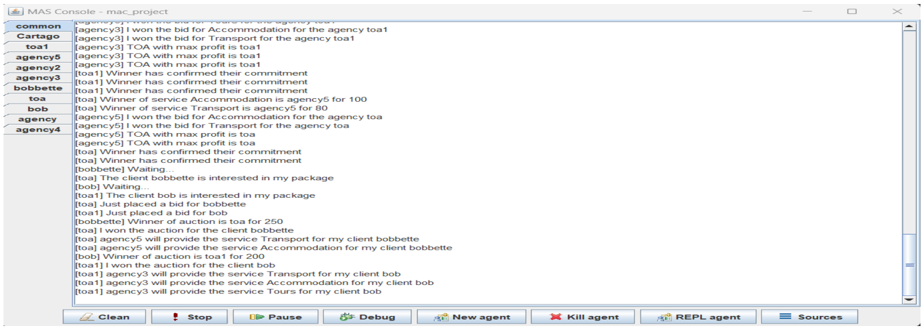
On the client side the agent will use the artifact to check if it is available to use, if it is not then it will wait until it is, and when it is available it will print out the message, wait the delay that is required and then print out another messaged announcing the end of that service. When running this phase, the goals do not wait for the completion of the previous ones, we did not have enough time to fix this, in the future fixing this would be a priority.

3 Running the code

Run Code To run the code, extract the zip file. Open the mac-project in the Intellij IDE. Then run the build.gradle file. You can also use the following command in the terminal, from inside the mac-project folder.

```
./gradlew -q --console=plain
```

This will start the first two phases automatically. Here is a sample output:



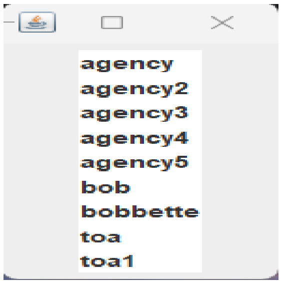
```
[agency3] I won the bid for Tours for the agency toa1
[agency3] I won the bid for Accommodation for the agency toa1
[agency3] I won the bid for Transport for the agency toa1
[agency3] TOA with max profit is toa1
[agency3] TOA with max profit is toa1
[agency3] TOA with max profit is toa1
[toa1] Winner has confirmed their commitment
[toa1] Winner has confirmed their commitment
[toa1] Winner has confirmed their commitment
[toa] Winner of service Accommodation is agency5 for 100
[toa] Winner of service Transport is agency5 for 80
[agency5] I won the bid for Accommodation for the agency toa
[agency5] I won the bid for Transport for the agency toa
[agency5] TOA with max profit is toa
[agency5] TOA with max profit is toa
[toa] Winner has confirmed their commitment
[toa] Winner has confirmed their commitment
[bobbette] Waiting...
[toa] The client bobbette is interested in my package
[bob] Waiting...
[toa1] The client bob is interested in my package
[toa] Just placed a bid for bobbette
[toa1] Just placed a bid for bob
[bobbette] Winner of auction is toa for 250
[toa] I won the auction for the client bobbette
[toa] agency5 will provide the service Transport for my client bobbette
[toa] agency5 will provide the service Accommodation for my client bobbette
[bob] Winner of auction is toa1 for 200
[toa1] I won the auction for the client bob
[toa1] agency3 will provide the service Transport for my client bob
[toa1] agency3 will provide the service Accommodation for my client bob
[toa1] agency3 will provide the service Tours for my client bob
```

Figure 4: Output after running the code

If the TOAs have successfully hired the service companies they needed and have been hired by a client you can then launch the third phase. In the REPL agent run a sample agent. Then give input

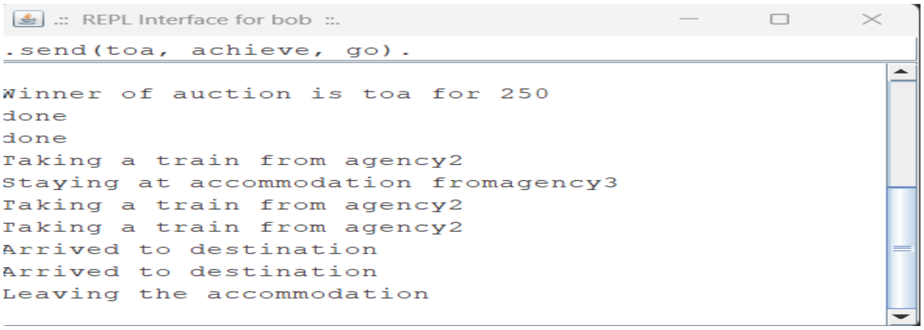
```
.send(toa, achieve, go).
```

This will start the third and final phase. We get the following output:



```
agency
agency2
agency3
agency4
agency5
bob
bobbette
toa
toa1
```

Figure 5: Agents

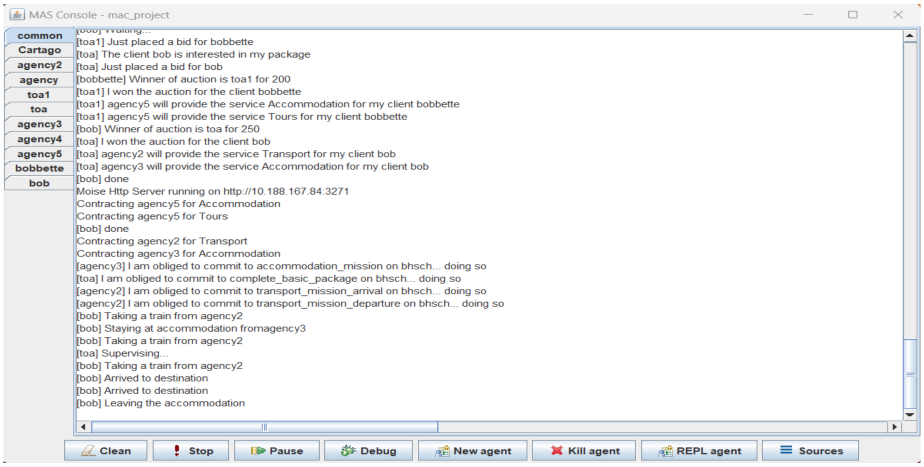


```
.send(toa, achieve, go).

Winner of auction is toa for 250
done
done
Taking a train from agency2
Staying at accommodation fromagency3
Taking a train from agency2
Taking a train from agency2
Arrived to destination
Arrived to destination
Leaving the accommodation
```

Figure 6: REPL interface for bob

We get the following output:



```
[bob] Waiting...
[toa1] Just placed a bid for bobbette
[toa] The client bob is interested in my package
[toa] Just placed a bid for bob
[bobbette] Winner of auction is toa1 for 200
[toa1] I won the auction for the client bobbette
[toa1] agency5 will provide the service Accommodation for my client bobbette
[toa1] agency5 will provide the service Tours for my client bobbette
[bob] Winner of auction is toa for 250
[toa] I won the auction for the client bob
[toa] agency2 will provide the service Transport for my client bob
[toa] agency3 will provide the service Accommodation for my client bob
[bob] done
Moise Http Server running on http://10.188.167.84:3271
Contracting agency5 for Accommodation
Contracting agency5 for Tours
[bob] done
Contracting agency2 for Transport
Contracting agency3 for Accommodation
[agency3] I am obliged to commit to accommodation_mission on bhsch... doing so
[toa] I am obliged to commit to complete_basic_package on bhsch... doing so
[agency2] I am obliged to commit to transport_mission_arrival on bhsch... doing so
[agency2] I am obliged to commit to transport_mission_departure on bhsch... doing so
[bob] Taking a train from agency2
[bob] Staying at accommodation fromagency3
[bob] Taking a train from agency2
[toa] Supervising...
[bob] Taking a train from agency2
[bob] Arrived to destination
[bob] Arrived to destination
[bob] Leaving the accommodation
```

Figure 7: Output after running the REPL interface for bob

Since some of the characteristics of certain agents are randomized it might take a few executions to get a scenario where both clients hire a TOA.

4 Final remarks

This project was a complex undertaking so even if we managed to accomplish the main guidelines, we also have ideas on how to make it better and more complete. In the future we could create a more complicated auction mechanism, right now the bidders have only one price that they can use to bid but it would be best to generate a price so that they can try to win the bid. We would also make the execution of the third phase automatic instead of using a message to trigger the plan. With more time we'd also implement the third type of package. Handling edge cases was something we began to do in the first phase but did not do in the third phase, we just have the execution fail in the clients don't find a tour operator that suits their needs. It would be interesting to try and develop a more intelligent way of solving this problem. Overall we have a code that works well in general and has the elements that were required for this project but there is surely space for improvement.