

Introduction

Diagram illustrating the Smart Museum platform architecture:

- Profiler Agent** (represented by a smartphone icon) interacts with the **Curator Agent** and the **Tour Guide Agent**.
- The **Curator Agent** is part of the **Exhibition** system, which includes the **Art Gallery** (represented by a cylinder icon).
- The **Tour Guide Agent** is part of the **Smart Museum platform**, which also includes a group of people viewing art.
- Arrows indicate the flow of information between the Profiler Agent, the Curator Agent, and the Tour Guide Agent.

From Nima Dokoohaki

Distributed AI and Intelligent Agents - Homework 1 - Page 1/6

The system is made of 3 agents that follow multiple behaviours, with the end goal being that the end-user gets a virtual tour of the art gallery tailored to his interests, directly on his smartphone.

Data model

Artefacts are of a certain genre (painting, statue, ...) and a certain category (classicism, romanticism, ...).

An Artefact also contains the following information:
ID, name, author, date of creation, place of creation.

A user profile is constituted of the following information:
Gender, Age, Occupation and Interests (a list of artefacts' categories).

Agents design

The agents that interact in this system are:

- The profiler agent
- The tour guide agent
- The curator agent

ProfilerAgent

The profiler agent contains some personal information (age, occupation, gender), the user's interests and the list of visited artefacts.

The profiler interacts with the tour guide to get a virtual tour (list of artefacts to visit) and the curator agent when "visiting" an artefact to get its detailed information.

TourGuideAgent

The tour guide interacts with the **CuratorAgent** to get artefacts suiting

the interests of a user:

When he gets a “get-tour” request from the **ProfilerAgent**, he builds a list of artifact’s (genre, category) pairs based on the user preferences, and he sends that list to the curator agent to get the ids of artifacts that verify those preferences.

CuratorAgent

The curator agent interacts with both the **TourGuideAgent** and the **ProfilerAgent**:

He gives a list of artifact ids to the **TourGuideAgent** when given a list of (genre, category) pairs by the **TourGuideAgent**, and gives an artifact’s detailed information when it’s requested by the **ProfilerAgent**.

Behaviours design

We had to implement the design described above, we used multiple behaviours for each one of the agents:

1. Profiler’s behaviours

- **RequestVirtualTourBehaviour** (extends **OneShotbehaviour**)

This behaviour sends a message to the **TourGuideAgent** (containing the user’s information: age, gender, interests, etc.) requesting a list of artifact IDs.

Once it gets this list, it creates the following sequential behavior:

- **Visiting sequential behaviours** (using a **SequentialBehaviour**)

We used a vanilla **SequentialBehaviour** where was added one **VisitingArtifactBehaviour** per artifact to visit, and one **NewTourBehaviour** at the end (to relaunch a new tour visit after a while)

- **VisitingArtifactBehaviour** (extends **OneShotbehaviour**)

This behaviour is linked with a given artifact ID, and sends a message to the **CuratorAgent** to get detailed information about that artifact.

Once that's done, the behaviour adds the artifact to the list of visited artifacts (if it wasn't previously visited) and then halts for a while (~2 seconds) to simulate the effect of a user browsing the artifacts one by one and taking time to read the information.

- **NewTourBehaviour** (extends **WakerBehaviour**)

This behaviour wakes after a certain time to request a new tour (by adding a **RequestVirtualTourBehaviour**) and is executed at the end of the sequential behaviour describe above. (after all artifacts were visited)

2. Tour-guide's behaviours

- **TourBuilderBehaviour** (extends **CyclicBehaviour**)

This behaviour builds a virtual tour by request of a **ProfilerAgent**. It does so by creating a list of a given size (in our case ~10) pairs of artifact category/genre, and sends them to the **CuratorAgent**. It then forwards the received artifact ids from the **CuratorAgent** to the **ProfilerAgent**.

- **Parallelized TourBuilderBehaviour** (with **ParallelBehaviour**)

Given the fact that we might have multiple users requesting a virtual guide from the same tour builder, we might want to process these in parallel instead of sequentially.

This is why we add a certain number of **TourBuilderBehaviour** behaviours to a **ParallelBehaviour**, and use it as our main behaviour.

3. Curator's behaviours

- **CuratorRequestsHandlingBehaviour** (extends **CyclicBehaviour**)

This is the main behaviour for our curator:

- 1) It answers a message from a **TourGuideAgent** with a set of artifact IDs that verify the given preferences.
- 2) It gives the detailed information of an artifact per request by a **ProfilerAgent**

Directory facilitator

1. Tour-Guide

Each Tour-Guide agent registers its TourBuilder service to the Directory Facilitator, identified by the agent name and the service type.

2. Profiler

a. Service research

Each Profiler agent start by asking the Directory Facilitator the list of every services matching a specified service type (here "TourGuideBuilder").

A request is then sent to the TourGuide agent selected by the user in order to retrieve a personalized tour.

b. Notification subscription

Each profiler agent subscribes to the Directory Facilitator in order to be notified by a SubscriptionInitiator behaviour when a new service of a specified type is published.