Xavier

Blanc

Agents IA pour les tests logiciel

11 JUIN 2024
BEFFROI DE MONTROUGE

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

CFTL
Comité Français des Tests Logiciels

Automatisation des tests

1. Définition des objectifs de test

- Identifier les fonctionnalités les plus critiques de l'application qui nécessitent des tests.
- Déterminer les types de tests à automatiser (tests unitaires, tests d'intégration, tests de performance, tests d'interface utilisateur, etc.).

2. Choix des outils et des technologies

- Sélectionner les outils d'automatisation de tests adaptés à l'application web et à son environnement de développement. Parmi les outils populaires, on trouve Selenium, Cypress, Puppeteer, et Playwright.
- Prendre en compte les langages de programmation supportés, la compatibilité avec les navigateurs, et l'intégration avec les systèmes de CI/CD (Intégration continue et Déploiement continu).

3. Planification des tests

- Établir un plan de test détaillé qui inclut les scénarios de test, les cas de test spécifiques, et les critères d'acceptation.
- Organiser les tests en suites de tests pour faciliter l'exécution et la maintenance.

4. Développement des scripts de test

- Écrire les scripts de test en utilisant l'outil d'automatisation choisi. Cela inclut la rédaction de fonctions pour interagir avec les éléments de l'interface utilisateur, simuler des actions d'utilisateurs, et vérifier les états de l'application.
- Utiliser des pratiques de développement logiciel comme la révision de code pour assurer la qualité des scripts de test.

5. Configuration de l'environnement de test

- Préparer l'environnement dans lequel les tests seront exécutés, ce qui peut inclure la configuration de serveurs de test, de bases de données de test, et d'autres services nécessaires.
- S'assurer que l'environnement de test est aussi proche que possible de l'environnement de production pour garantir la fiabilité des résultats de test.

6. Exécution des tests et rapports

- Exécuter les tests automatiquement selon un calendrier (par exemple, à chaque push dans le dépôt de code) ou sur demande.
- Configurer la génération de rapports pour collecter les résultats des tests, identifier les échecs, et analyser les problèmes.

7. Maintenance des scripts de test

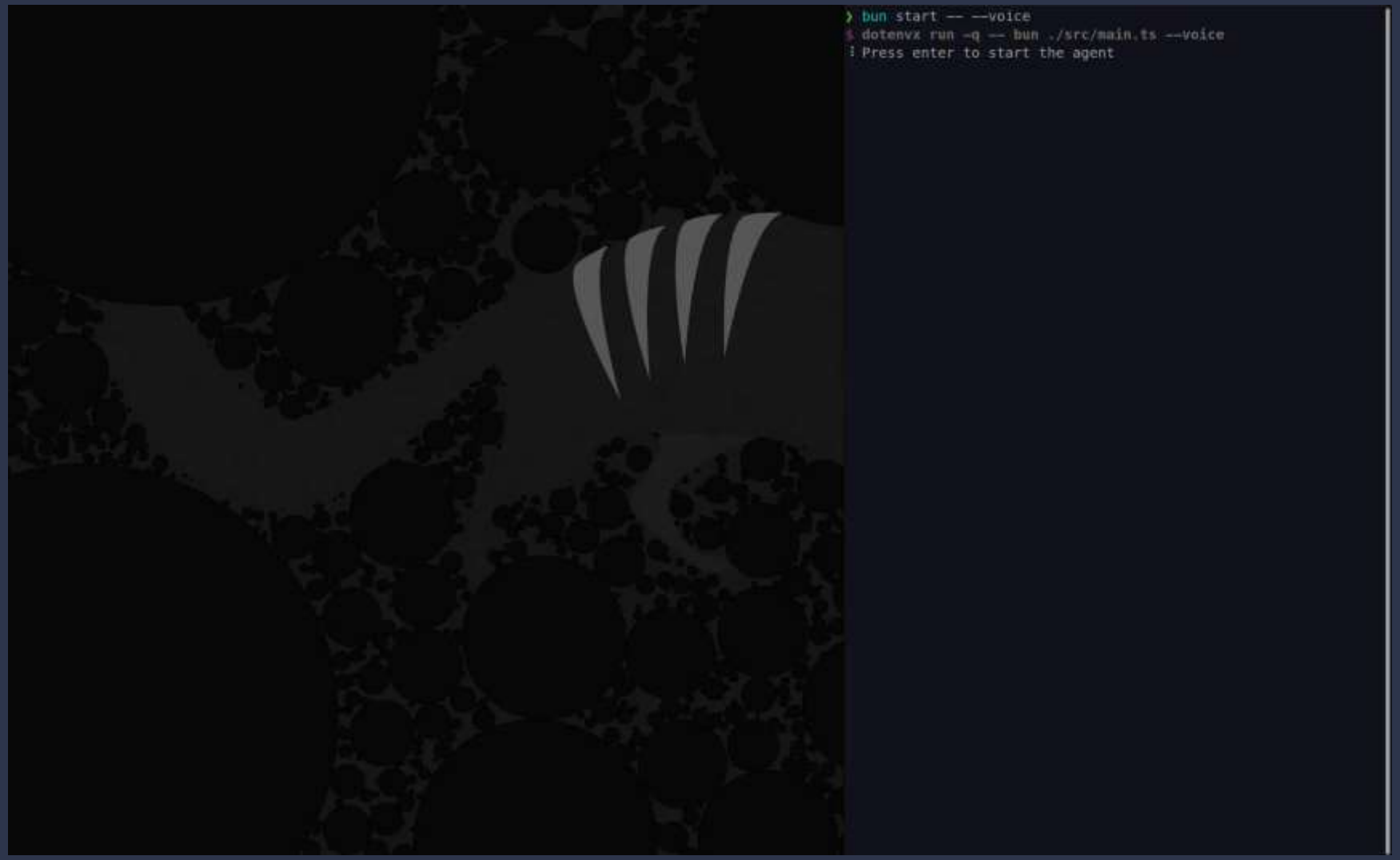
- Mettre régulièrement à jour les scripts de test pour refléter les changements dans l'application web.
- Optimiser et refactoriser les scripts de test pour améliorer leur efficacité et leur maintenabilité.

8. Intégration continue

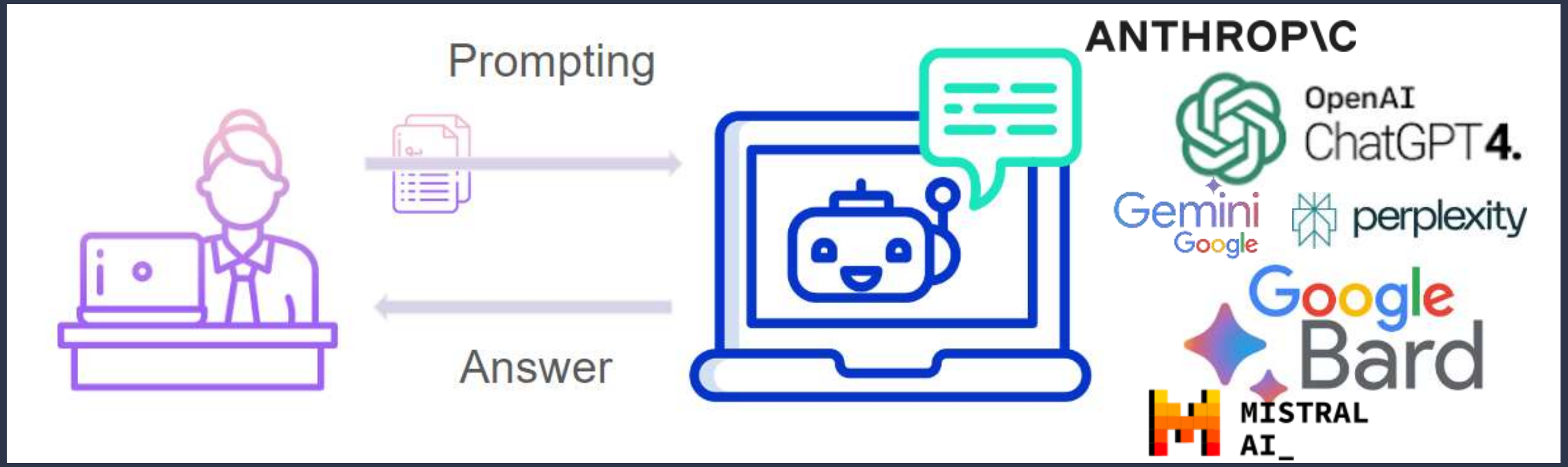
- Intégrer l'automatisation des tests dans le pipeline CI/CD pour que les tests soient exécutés automatiquement à chaque modification du code source.

Quelle place pour l’IA ?

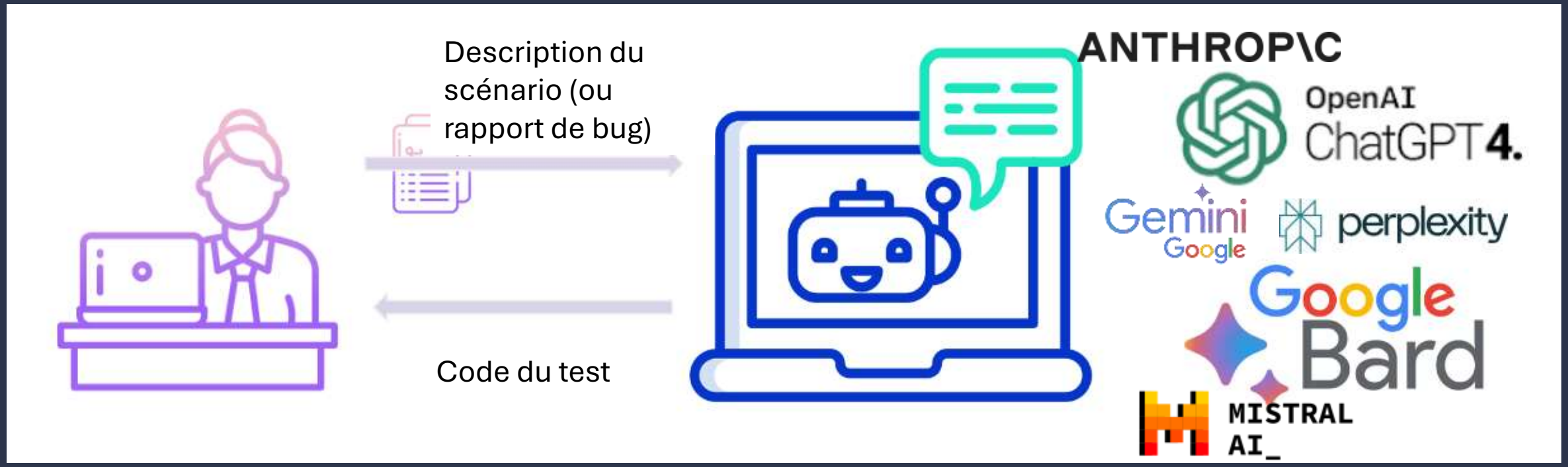
Démo


bun start -- --voice
dotenvx run -q -- bun ./src/main.ts --voice
Press enter to start the agent

Les LLMs


Prompting
Answer
ANTHROP\C
OpenAI
ChatGPT 4.
Gemini
Google
perplexity
Google
Bard
MISTRAL
AI_

Ce qui marche (mais pas à 100%)



Exemple avec ChatGPT 4

Voici un test d'une application web.

1 Recharger la page courante
2 Rechercher un véhicule ayant les caractéristiques suivantes :
3 choisir la valeur **BMW** pour **Marque**
4 choisir la valeur **M2** pour **Modèle**
5 choisir la valeur **Diesel** pour **Carburation**
6 cocher la case **Nouveau véhicule**
7 Saisir les informations suivantes sur conducteur :
8 entrer le texte **Aurelia Vaughan** dans **Nom**
9 Entrer la date **03/08/1990** dans **Date d'obtention**
10 entrer le texte **75653** dans **Code Postal**
11 entrer le texte **-48** dans **Bonus/Malus**
12 Choisir l'option **Tous risques**
13 Cliquer sur **simuler**

A la fin, il faut Vérifier l'existence du texte Tarif annuel : 4155.84

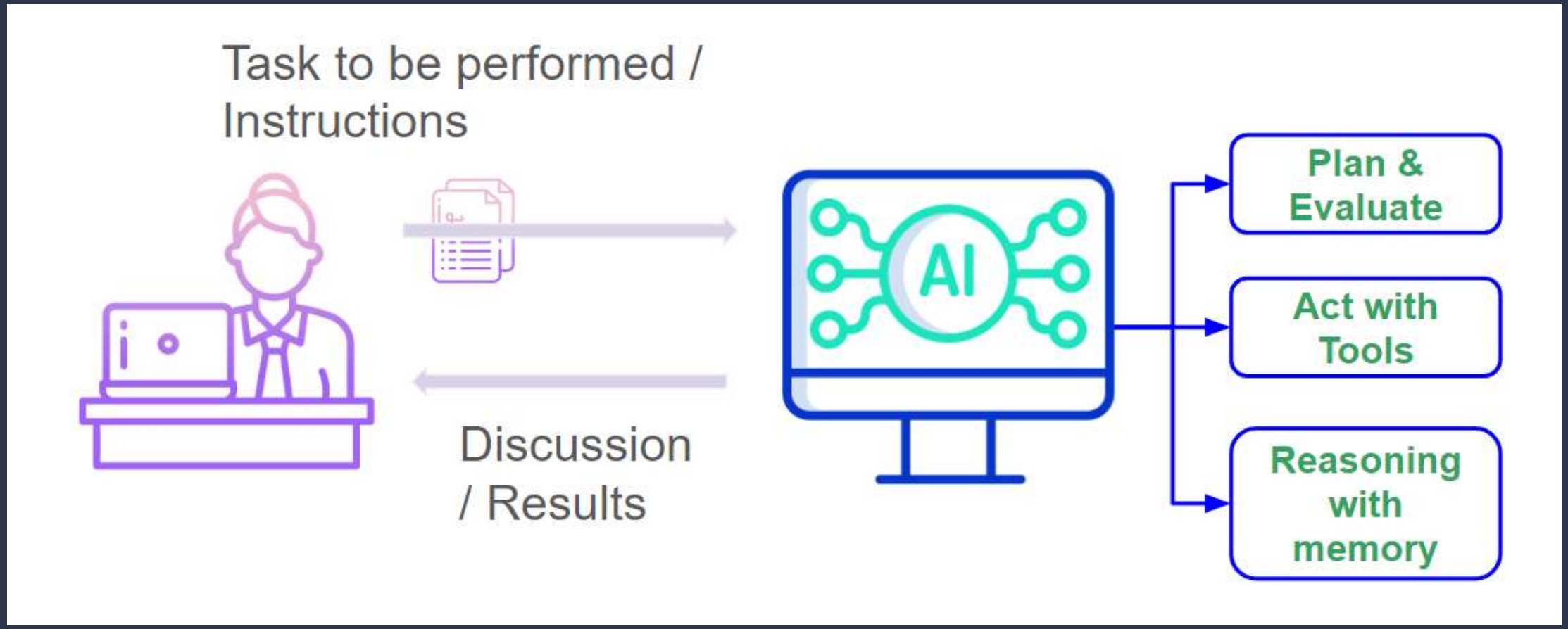
Peux-tu me donner le code playwright ?

javascript Copy code

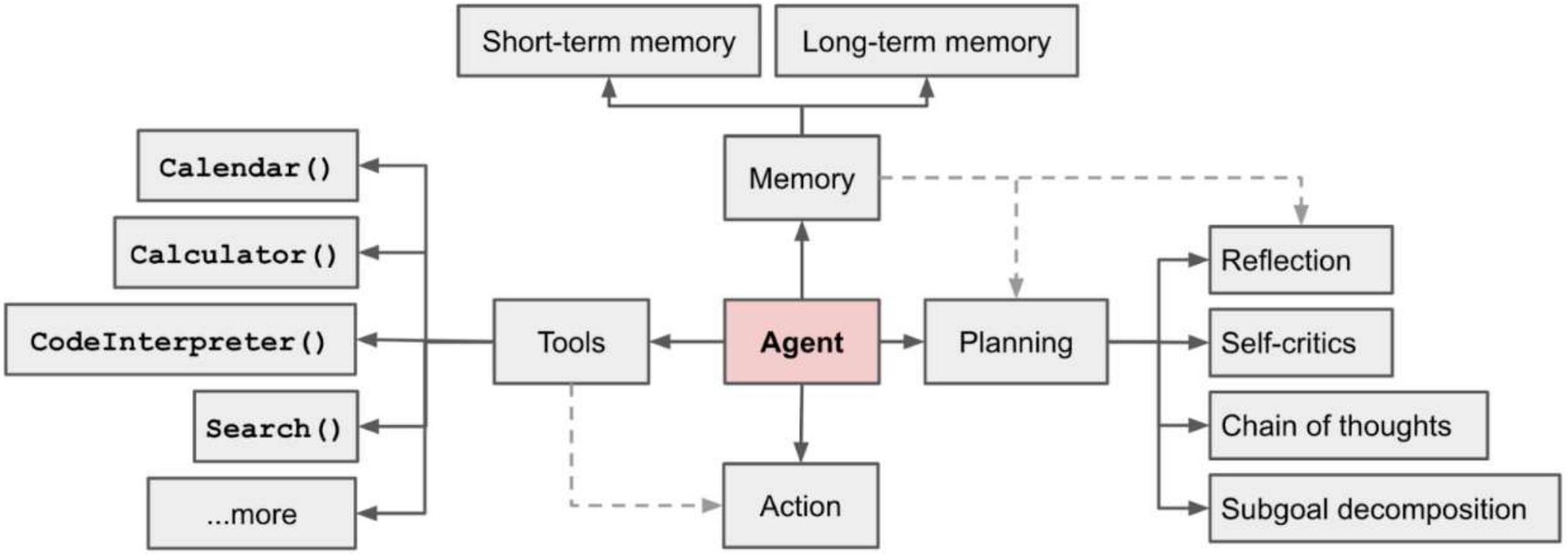
```javascript
const { firefox } = require('playwright'); // ou bien 'chromium' ou 'webkit'

(async () => {
  const browser = await firefox.launch({ headless: false }); // Mettez 'headless: tru
  const page = await browser.newPage();
  await page.goto('https://example.com'); // Remplacez 'https://example.com' par l'UR

  // 1. Recharger la page courante
  await page.reload();

  // 2-6. Rechercher un véhicule avec les caractéristiques données
  await page.selectOption('select#marque', 'BMW'); // Remplacez 'select#marque' par l
  await page.selectOption('select#modele', 'M2'); // Remplacez 'select#modele' par le
  await page.selectOption('select#carburation', 'Diesel'); // Remplacez 'select#carbu
  await page.check('input#nouveauVehicule'); // Remplacez 'input#nouveauVehicule' par

  // 7-12. Saisir les informations sur le conducteur
  await page.fill('input#nom', 'Aurelia Vaughan'); // Remplacez 'input#nom' par le sé
  await page.fill('input#dateObtention', '03/08/1990'); // Remplacez 'input#dateObten
  await page.fill('input#codePostal', '75653'); // Remplacez 'input#codePostal' par l
  await page.fill('input#bonusMalus', '-48'); // Remplacez 'input#bonusMalus' par le
  await page.selectOption('select#couverture', 'Tous risques'); // Remplacez 'select#

  // 13. Cliquer sur simuler
  await page.click('button#simuler'); // Remplacez 'button#simuler' par le sélecteur

  // Vérification de l'existence d xte "Tarif annuel : 4155.84"
  await page.waitForSelector('text="Tarif annuel : 4155.84"');
```

Remplacez les sélecteurs dans le code (comme `'select#marque'`, `'input#nom'`, etc.) par les vrais sélecteurs présents dans votre application web. Les sélecteurs dépendent de la structure HTML de votre site. Utilisez les outils de développement de votre navigateur pour les trouver.

Des limites


Task to be performed /
Instructions
Discussion
/ Results
AI
Plan &
Evaluate
Act with
Tools
Reasoning
with
memory

Les Agents IA qui utilisent des LLMs



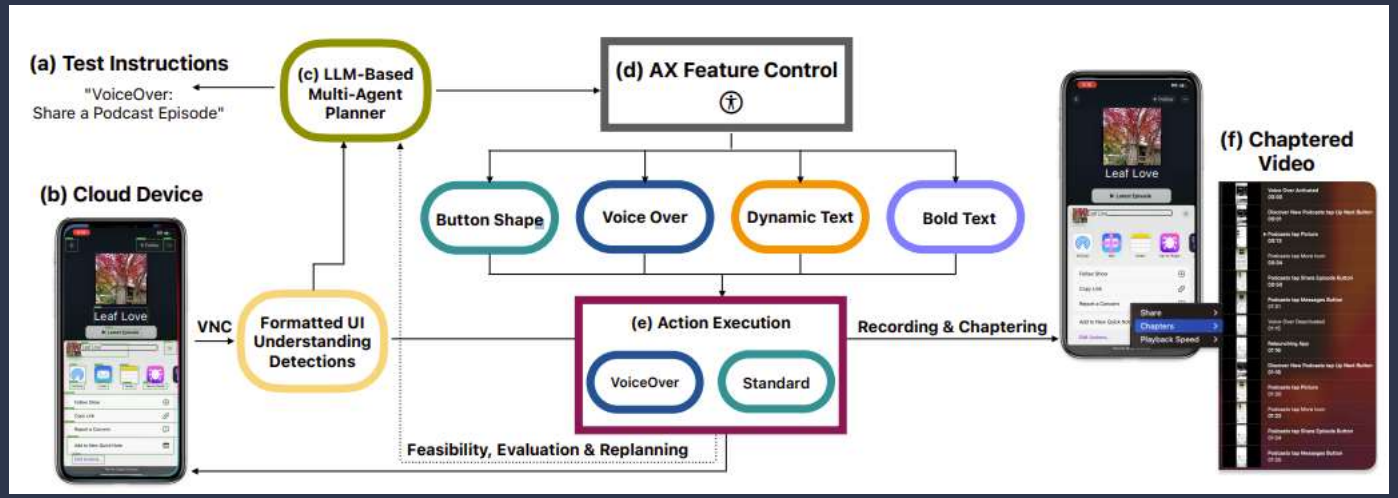
Source: Weng, Lilian. (Jun 2023). LLM-powered Autonomous Agents". Lil'Log. https://lilianweng.github.io/posts/2023-06-23-agent/ .

Agents Testeurs : deux exemples

- AXNav – Agent de tests d’accessibilité (Apple Labs) :
  - Rejoue les tests d'accessibilité à partir d’une consigne exprimée en langage naturelle
  - Vise 4 caractéristiques d’accessibilité avec exécution des tests sur des app mobiles IoS

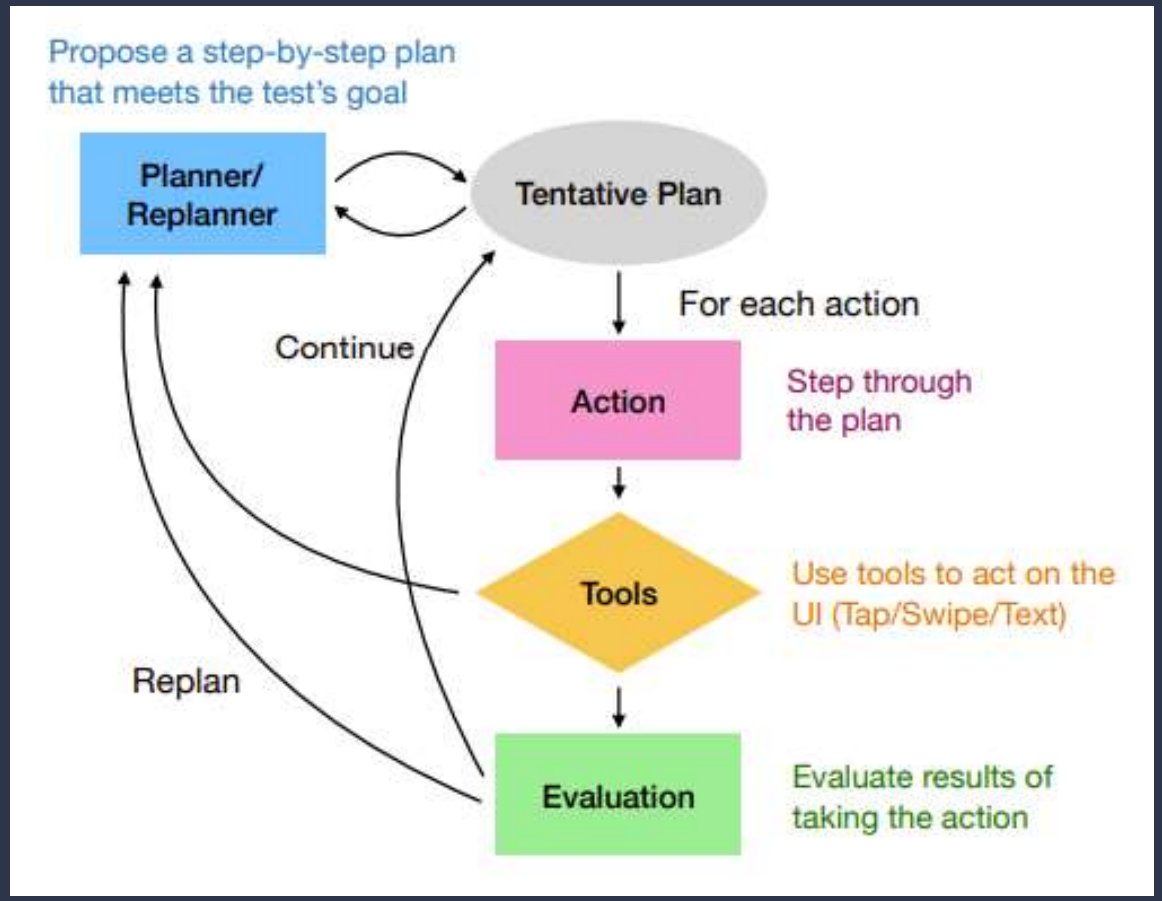
- Gérald - Agent Testeur Manuel (Smartesting AI Labs) :
  - Prototype d’agent Testeur manuel pour les applications web à partir d’une description en langage naturel des cas de test à exécuter

AXNav - Testeur d’accessibilité

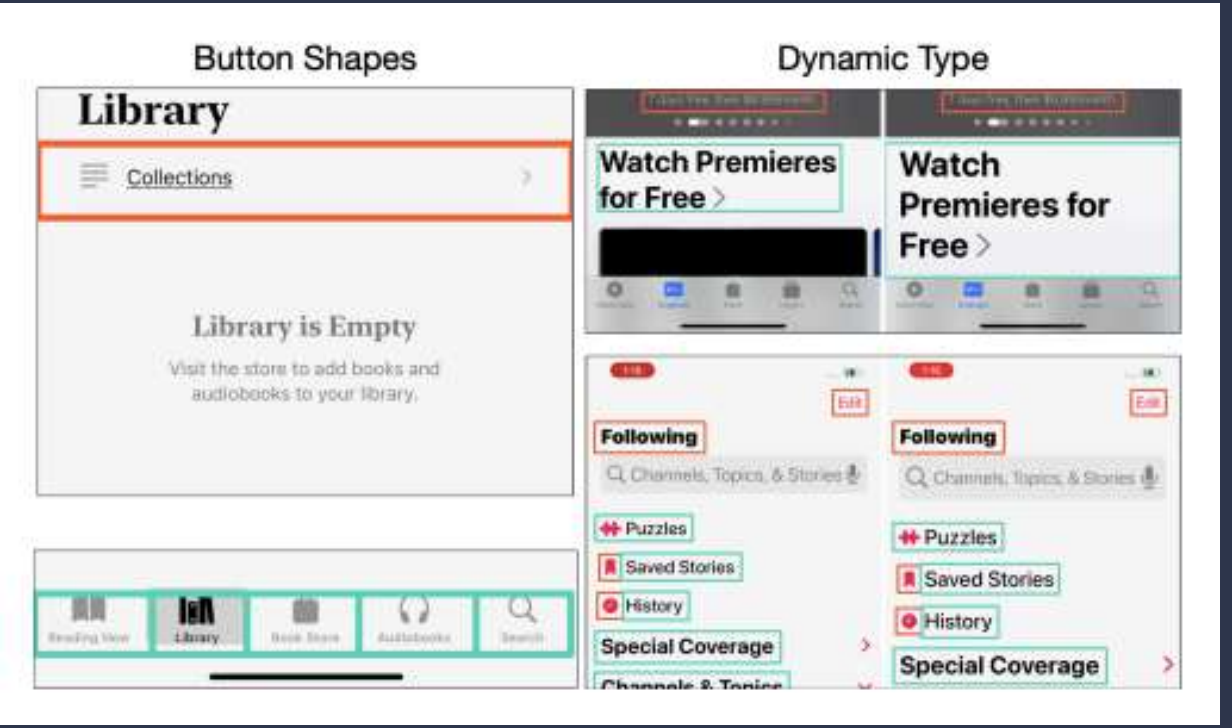


Source : AXNav: Replaying Accessibility Tests from Natural Language - https://arxiv.org/abs/2310.02424 - 5/3/24

AXNav - Planification



Reporting


Button Shapes
Dynamic Type
Library
Collections
Library is Empty
Visit the store to add books and audiobooks to your library.
Watch Premieres for Free
Watch Premieres for Free
Following
Puzzles
Saved Stories
History
Special Coverage
Following
Puzzles
Saved Stories
History
Special Coverage

Résultats AXNav

- Résultats tels qu’indiqués dans l’article :
  - en testant avec des instructions manuelles réalistes, un taux de réussite de 70 % et 85 % pour la navigation est obtenu.
  - le système a été évalué auprès de 10 testeurs d'accessibilité professionnels qui ont trouvé ce système très utile dans leur travail.

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Gérald - Agent Testeur Manuel

▶ Expérimenté-Paris-MALUS-Assurance tous risques-TARIF

#	Actions	Résultats attendus
1	Recharger la page courante	
2	Rechercher un véhicule ayant les caractèristiques suivantes :	
3	choisir la valeur BMW pour Marque	
4	choisir la valeur M2 pour Modèle	
5	choisir la valeur Diesel pour Carburation	
6	cocher la case Nouveau véhicule	
7	Saisir les informations suivantes sur conducteur :	
8	entrer le texte Aurelia Vaughan dans Nom	
9	Entrer la date 03/08/1990 dans Date d'obtention	
10	entrer le texte 75653 dans Code Postal	
11	entrer le texte -48 dans Bonus/Malus	
12	Choisir l'option Tous risques	
13	Cliquer sur simuler	Vérifier l'existence du texte Tarif annuel : 4155.84

Description du test en langage naturel

Application disponible pour exécuter des tests

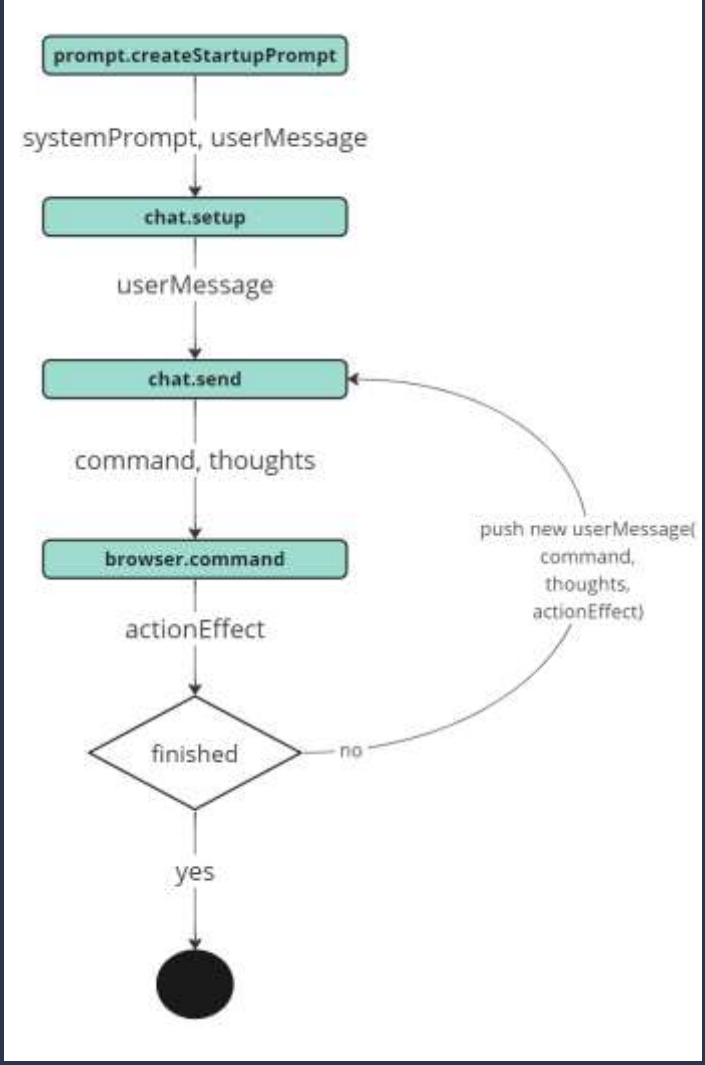
Générateur d'action

Evaluateur

Exécuteur d'action

Web Driver

Boucle rétroactive (infinie)



Création du system prompt et du premier message

(le premier message contient le scénario de test)

Boucle principale :

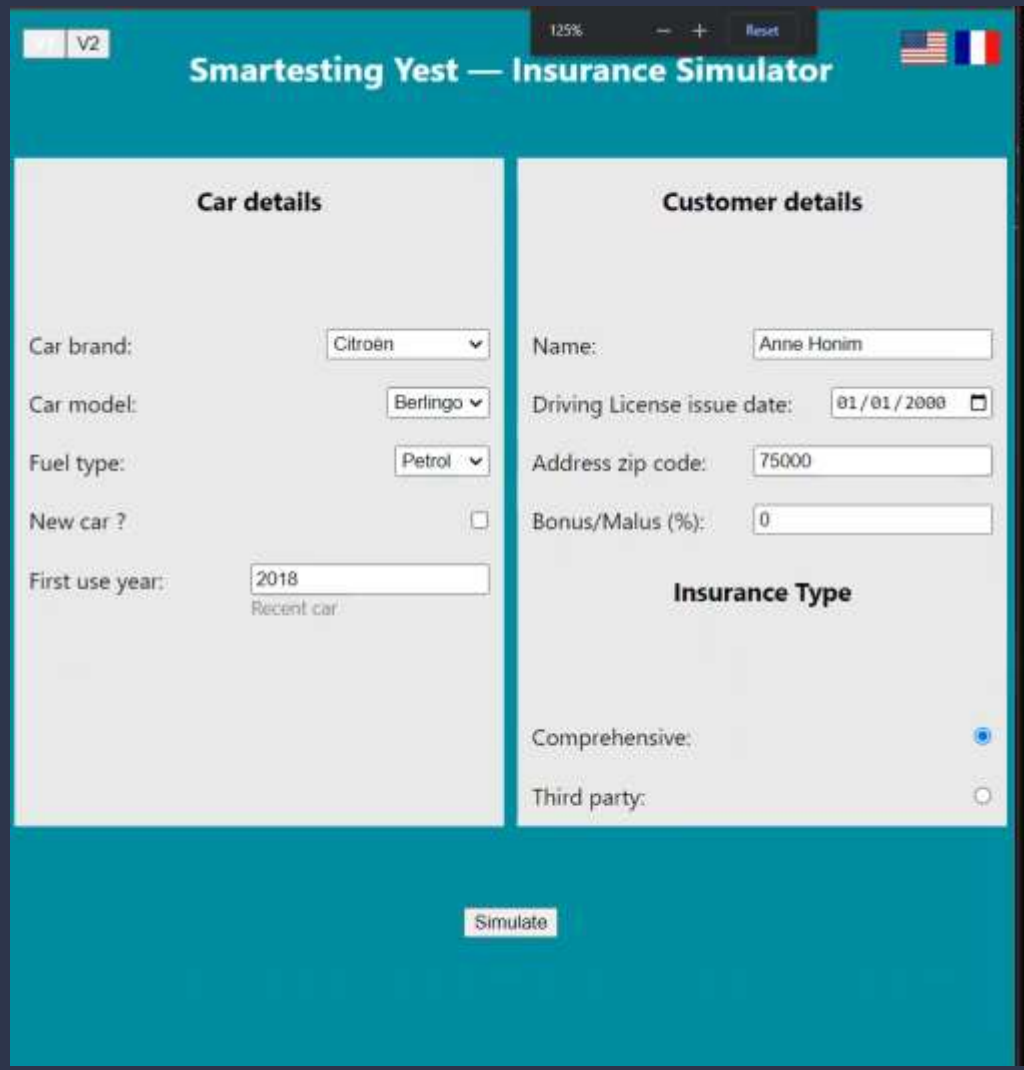
1.réponse du LLM
2.extraction de la commande
3.exécution de la commande
4.envoi du retour de la commande au LLM

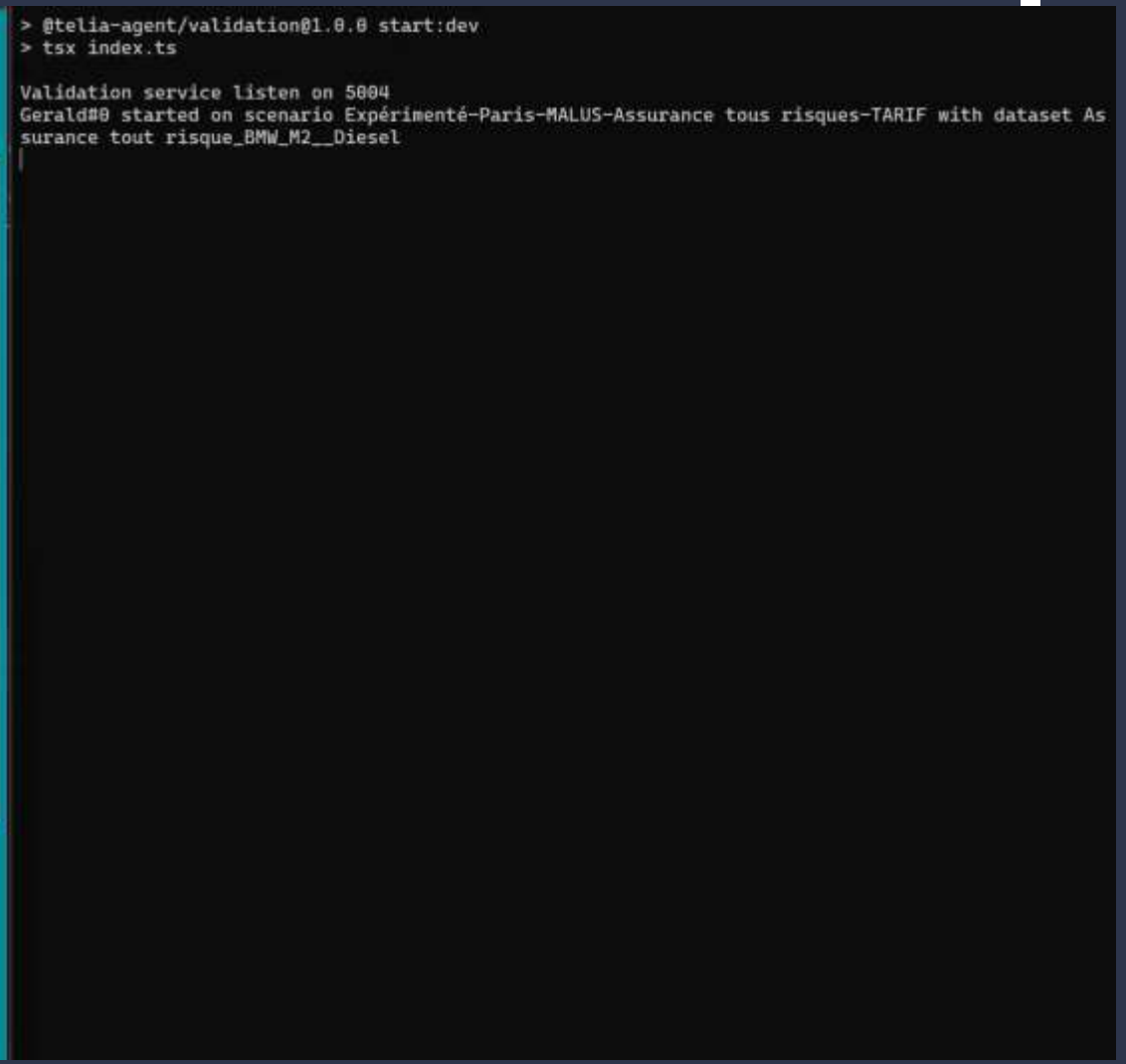
L'exécution prend fin à la réception de la commande "finish"

Exemple de description de test (yest)

Expérimenté-Paris-MALUS-Assurance tous risques-TARIF

#	Actions	Résultats attendus
1	Recharger la page courante	
2	Rechercher un véhicule ayant les caractèristiques suivantes :	
3	choisir la valeur BMW pour Marque	
4	choisir la valeur M2 pour Modèle	
5	choisir la valeur Diesel pour Carburation	
6	cocher la case Nouveau véhicule	
7	Saisir les informations suivantes sur conducteur :	
8	entrer le texte Aurelia Vaughan dans Nom	
9	Entrer la date 03/08/1990 dans Date d'obtention	
10	entrer le texte 75653 dans Code Postal	
11	entrer le texte -48 dans Bonus/Malus	
12	Choisir l'option Tous risques	
13	Cliquer sur simuler	Vérifier l'existence du texte Tarif annuel : 4155.84


V2
125%
Reset
Smartesting Yest — Insurance Simulator
Car details
Car brand:
Citroën
Car model:
Berlingo
Fuel type:
Petrol
New car ?
First use year:
2018
Recent car
Customer details
Name:
Anne Honim
Driving License issue date:
01/01/2000
Address zip code:
75000
Bonus/Malus (%):
0
Insurance Type
Comprehensive:
Third party:
Simulate


> @telia-agent/validation@1.0.0 start:dev
> tsx index.ts
Validation service listen on 5004
Gerald#0 started on scenario Expérimenté-Paris-MALUS-Assurance tous risques-TARIF with dataset Assurance tout risque_BMW_M2__Diesel

Résultats Gérald

- Résultats évaluation :
  - On arrive à 100% du scénario
  - Mais des itérations plus ou moins longue
  - Problème de précision des actions
  - Problème de la taille de la mémoire

- Feuille de route :
  - Poursuivre la mise au point sur l'application "Simulateur Assurance"
  - Etendre les essais à d'autres applications contrôlées en Laboratoire
  - Passer en évaluation avec des utilisateurs de Yest
  - Étendre l'agent pour qu'il prenne en entrée une plus grande variété de cas de tests manuels

Discussion

- l’architecture des Agents LLM ouvre la voie vers des traitements autonomes
- la capacité d’évaluer les résultats est clé
- La mise au point reste complexe

State Of The Art

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Front. Comput. Sci., 2024, 18(6): 186345
https://doi.org/10.1007/s11704-024-40231-1

REVIEW ARTICLE

A survey on large language model based aut

Lei WANG, Chen MA*, Xueyang FENG*, Zeyu ZHANG, Hao Y
Zhiyuan CHEN, Jiakai TANG, Xu CHEN (✉), Yanka
Wayne Xin ZHAO, Zhewei WEI, Jirong WE
Gaoling School of Artificial Intelligence, Renmin University of China, Be

Abstract Autonomous agents have long been a research focus in academic and industry communities. Previous research often focuses on training agents with limited knowledge within isolated environments, which diverges significantly from human learning processes, and makes the agents hard to achieve human-like decisions. Recently, through the acquisition of vast amounts of Web knowledge, large language models (LLMs) have shown potential in human-level intelligence, leading to a surge in research on LLM-based autonomous agents. In this paper, we present a comprehensive survey of these studies, delivering a systematic review of LLM-based autonomous agents from a holistic perspective. We first discuss learning process, since t and individuals can lea environments. Because of the previous studies are level decision processes, domain settings.

In recent years, larg achieved notable succ potential in attaining hu capability arises from datasets alongside a subs Building upon this cap

Preprint

VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks

Jing Yu Koh Robert Lo* Lawrence Jang* Vikram Duvvur*
Ming Chong Lim* Po-Yu Huang* Graham Neubig Shuyan Zhou
Ruslan Salakhutdinov Daniel Fried
Carnegie Mellon University
{jingyuk,rsalakhu,dfried}@cs.cmu.edu

Abstract

Autonomous agents capable of planning, reasoning, and executing web offer a promising avenue for automating computer tasks. How jority of existing benchmarks primarily focus on text-based agen many natural tasks that require visual information to effectively that most computer interfaces cater to human perception, visual in ten augments textual data in ways that text-only models struggle fectively. To bridge this gap, we introduce VisualWebArena, a b signed to assess the performance of multimodal web agents on rea *grounded tasks*. VisualWebArena comprises of a set of diverse and based tasks that evaluate various capabilities of autonomous multi To perform on this benchmark, agents need to accurately process

GPT-4V(ision) is a Generalist Web Agent, if Grounded

Boyuan Zheng[1] Boyu Gou[1] Jihyung Kil[1] Huan Sun[1] Yu Su[1]
https://osu-nlp-group.github.io/SeeAct

Abstract

The recent development on large multimodal models (LMMs), especially GPT-4V(ision) and Gemini, has been quickly expanding the capability boundaries of multimodal models beyond traditional tasks like image captioning and visual question answering. In this work, we explore the potential of LMMs like GPT-4V as a generalist web agent that can follow natural language instructions to complete tasks on any given website. We propose SeeAct, a generalist web agent that harnesses the power of LMMs for integrated visual understanding and acting on the web. We evaluate on the recent Mind2Web benchmark. In addition to standard offline evaluation on cached websites, we enable a new online evaluation setting by developing a tool that allows running web agents on live websites. We show that GPT-4V presents a great potential for web agents—it can successfully complete 51.1% of the tasks on live websites if we manually ground its textual plans into actions on

Figure 1: SeeAct leverages an LMM like GPT-4V to visually perceive websites and generate plans in textual forms. The textual plans are then grounded onto the HTML elements and operations to act on the website.

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Quel impact pour les testeurs ?

Généré avec Dall-E par Bruno Legeard en novembre 2023

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

VOTEZ POUR LA MEILLEURE PRÉSENTATION

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Xavier

Blanc

Agents IA pour les tests logiciel

11 JUIN 2024
BEFFROI DE MONTROUGE

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

CFTL
Comité Français des Tests Logiciels