Integrarea AI-ului intr-o aplicatie informatica

1. Inteligenta artificiala nu inseamna numai conversatie

Majoritatea utilizatorilor cunosc inteligenta artificiala prin aplicatii precum ChatGPT, Gemini sau Claude. Acestea permit formularea unor intrebari si primirea unor raspunsuri, prin text sau voce.

Dar inteligenta artificiala poate fi folosita si direct din interiorul unei aplicatii informatice.

In aceasta situatie, nu neaparat utilizatorul este cel care comunica cu AI-ul. Aplicatia transmite solicitarile, primeste rezultatele si le foloseste in propriile prelucrari.

Aceasta diferenta este esentiala. Nu discutam despre deschiderea unei ferestre de conversatie, ci despre integrarea unor servicii de inteligenta artificiala in functionarea unei aplicatii.

2. Unde se afla inteligenta artificiala?

Serviciile oferite de OpenAI, Google sau Anthropic functioneaza pe calculatoare administrate de furnizori, accesibile prin internet.

Acest mod de furnizare a resurselor informatice la distanta este cunoscut sub denumirea de *cloud*.

Aplicatia utilizatorului nu trebuie sa contina modelul AI. Ea poate solicita prelucrari unui serviciu extern (AI) si poate primi rezultatele acestora.

Exista si modele AI care pot functiona local, pe calculatorul utilizatorului. Acestea reprezinta o alta modalitate de integrare, care nu constituie obiectul principal al articolului.

3. Cheia de acces si mecanismul de comunicare

Pentru a utiliza un serviciu AI dintr-o aplicatie, dezvoltatorul trebuie sa obtina acces la serviciul respectiv.

Furnizorul pune la dispozitie o interfata de comunicare numita API, prescurtare de la *Application Programming Interface*.

API-ul reprezinta un ansamblu de reguli prin care o aplicatie solicita servicii altei aplicatii si primeste rezultatele.

Accesul se realizeaza, de regula, folosind o cheie API, o combinatie de caractere asociata unui cont sau proiect, comparabila cu o parola de acces.

Cheia permite identificarea accesului autorizat si contabilizarea utilizarii serviciului. Ea trebuie pastrata in siguranta si nu trebuie expusa. In etapa de realizare si testare, dezvoltatorul poate utiliza propria cheie API. In exploatarea unei aplicatii distribuite mai multor beneficiari, in modelul prezentat aici, fiecare beneficiar foloseste propria cheie API si suporta direct costul prelucrarilor sale.

Abonamentul pentru utilizarea unei aplicatii de conversatie nu trebuie confundat cu accesul la serviciile API, care au conditii de lucru si costuri separate.

4. Cum functioneaza efectiv o apelare AI?

Utilizarea AI prin API poate fi comparata cu solicitarea unui serviciu la un ghiseu.

Aplicatia transmite o cerere, serviciul AI efectueaza prelucrarea si returneaza rezultatul. Operatiunea se incheie si costurile sunt inregistrate.

O alta comparatie este convorbirea telefonica platita in functie de consum. Fiecare apel reprezinta o noua comunicare si poate genera un cost.

In utilizarea obisnuita prin API, AI-ul nu ramane permanent activ in interiorul aplicatiei. Fiecare solicitare determina o prelucrare distincta. In cazul apelurilor independente, AI-ul nu primeste automat istoricul cererilor anterioare.

Daca noua cerere depinde de activitatea anterioara, trebuie asigurata transmiterea contextului necesar, inclusiv stadiul la care a ramas activitatea. Aceasta se poate realiza prin informatiile furnizate de aplicatie sau prin mecanismele de continuitate oferite de serviciul AI.

Aplicatia ramane responsabila de organizarea activitatii si de pastrarea rezultatelor furnizate de AI. API-ul este interfata tehnica prin care aplicatia comunica cu serviciul AI.

Daca aplicatia este interactiva, cerintele pot fi introduse printr-o interfata de tip chat, prin voce sau prin alte comenzi. Utilizatorul poate formula sarcini in limbaj obisnuit, iar AI-ul returneaza raspunsul aplicatiei. Aceasta decide ce rezultat afiseaza utilizatorului si in ce forma.

5. Cum poate aplicatia sa transmita comenzi catre AI?

O cerere poate fi initiata in mai multe moduri.

Utilizatorul poate introduce o instructiune prin tastatura sau o poate dicta. Dar aplicatia poate transmite si comenzi pregatite anterior, fara ca utilizatorul sa formuleze personal cererea.

De exemplu, o aplicatie poate avea un buton numit „Genereaza structura proiectului”.

La apasarea butonului, aplicatia construieste si transmite catre AI o cerere prestabilita, insotita de informatiile necesare.

O asemenea instructiune transmisa modelului AI este numita frecvent *prompt*.

Comanda poate fi initiata si automat, atunci cand aplicatia ajunge la o anumita etapa a prelucrarii. Prin urmare, utilizarea AI nu presupune obligatoriu o conversatie interactiva.

6. Comunicarea functioneaza in ambele sensuri

Aplicatia nu este limitata la transmiterea unei cereri si asteptarea raspunsului final.

In timpul prelucrarii, AI-ul poate avea nevoie de informatii suplimentare. El poate cere lamuriri utilizatorului prin intermediul aplicatiei sau poate solicita apelarea unor functii puse la dispozitie de aceasta.

Pentru aceasta, aplicatia pune la dispozitie anumite functii, descriind numele lor, parametrii acceptati si rezultatele furnizate. Aceste descrieri formeaza o conventie tehnica de comunicare intre AI si aplicatie.

AI-ul poate solicita aplicatiei sa execute una dintre aceste functii, indicand parametrii necesari.

De exemplu, poate solicita:

CitesteDocument(codDocument)

AI-ul nu executa direct aceasta functie in aplicatia utilizatorului. El transmite o solicitare de apelare.

Aplicatia verifica solicitarea, executa functia daca este permisa si transmite rezultatul catre AI.

Acest mecanism este cunoscut sub denumirea de *function calling* sau *tool calling*.

El permite AI-ului sa solicite informatii sau operatii suplimentare fara sa primeasca acces direct si necontrolat la baza de date ori la fisierele aplicatiei.

7. Cum sunt primite si pastrate rezultatele?

Dupa prelucrare, serviciul AI returneaza aplicatiei un raspuns.

Acesta poate contine un text, o structura de date sau alte tipuri de rezultate pe care serviciul AI le pune la dispozitia aplicatiei care l-a apelat.

Pentru informatii simple, rezultatul poate fi preluat de aplicatie direct in variabile sau structuri de memorie.

Pentru informatii complexe ori voluminoase, aplicatia poate crea fisiere intermediare, numite si fisiere de manevra, pe care le poate folosi ulterior.

Aplicatia este cea care primeste datele, le verifica, le organizeaza si le scrie in fisierele sale. In aceasta arhitectura, AI-ul nu are acces direct la dispozitivele de stocare ale beneficiarului.

De exemplu, AI-ul poate genera o structura ierarhica a unui proiect, cu capitole, subcapitole si descrieri.

Aplicatia primeste structura, o poate salva intr-un fisier text si, dupa verificare si eventuale modificari, o importa in propriul sistem de baza de date.

Ulterior, o noua comanda poate solicita AI-ului traducerea structurii in limba franceza. Aplicatia primeste traducerea si poate genera un document DOCX folosind functiile sale de editare si export.

Astfel, AI-ul realizeaza prelucrarea solicitata, iar aplicatia gestioneaza rezultatul si documentele finale.

8. Contextul: ce trebuie sa stie AI-ul despre activitatea sa?

Un model AI nu cunoaste automat proiectul in care este utilizat.

Pentru a realiza corect o prelucrare, trebuie sa primeasca informatiile necesare.

Aceste informatii formeaza *contextul*.

Putem distinge doua categorii:

Contextul general cuprinde informatii despre sistemul informatic, rolul AI-ului, regulile de lucru si functiile disponibile.

Contextul operational cuprinde informatiile despre sarcina concreta: ce trebuie facut acum, datele si documentele implicate si stadiul la care a ramas activitatea.

Pentru organizarea colaborarii se pot defini protocoale de lucru: reguli, descrieri de situatii, termeni si instructiuni formulate intr-un limbaj inteligibil. Prin acestea, AI-ul afla in ce mediu lucreaza, ce reguli trebuie sa respecte si cum poate aborda sarcinile primite.

De exemplu, in metodologia MAP au fost definite protocoalele AI-Ready si QMAP, destinate pregatirii contextului si organizarii colaborarii dintre aplicatie si AI; vezi cartea MAP.

Aceste protocoale sunt reguli stabilite pentru aplicatia respectiva, nu cerinte obligatorii ale furnizorilor de servicii AI.

Pentru activitati desfasurate in mai multe etape, pregatirea si administrarea contextului (punerea in tema a AI-ului) devin insa foarte importante.

9. De ce trebuie refacut sau recuperat contextul?

In modul obisnuit de apelare independenta prin API, modelul AI nu primeste automat informatiile din solicitarile anterioare.

La o noua apelare, el trebuie pus in tema cu ceea ce este necesar pentru continuarea activitatii (daca trebuie continuata activitatea anterioara).

Aplicatia poate transmite din nou instructiunile, documentele si rezultatele relevante chiar daca acele rezultate au fost obtinute tot de AI intr-o apelare anterioara.

Unele servicii AI ofera si mecanisme de pastrare a contextului intre apelari, dar acestea nu elimina responsabilitatea aplicatiei de a asigura informatiile actuale si necesare fiecarei prelucrari.

De aceea, pentru sistemele informatice complexe, este recomandabil ca aplicatia sa pastreze independent informatiile esentiale pentru continuarea activitatii.

10. Utilizarea interactiva si utilizarea automata

Integrarea AI poate fi realizata atat in aplicatii interactive, cat si in aplicatii cu prelucrare automata (batch).

Intr-o aplicatie interactiva, utilizatorul poate declansa o operatie printr-un buton, poate introduce o cerere sau poate participa la o conversatie.

Intr-o prelucrare batch, cererile catre AI sunt initiate automat de aplicatie, fara interventia operatorului.

In ambele cazuri, mecanismul de baza este acelasi: aplicatia pregateste solicitarea, o transmite serviciului AI, primeste raspunsul si continua prelucrarea.

In cazul aplicatiilor batch, trebuie prevazute si situatiile in care serviciul nu raspunde, apar erori sau este necesara reluarea unei operatii.

La reluare, aplicatia trebuie sa stie ce s-a executat deja, ca aceeasi operatie sa nu fie facuta de doua ori.

Nu orice prelucrare automata necesita AI. Daca operatia poate fi realizata mai simplu si mai sigur prin algoritmi obisnuiti, utilizarea AI poate fi inutila.

11. Costul prelucrarilor si optimizarea consumului

Utilizarea serviciilor AI prin API genereaza costuri in functie de consumul efectuat in exploatarea aplicatiei.

Ce este un token si cat costa?

Un token este o unitate folosita de modelele AI pentru prelucrarea textului. El poate reprezenta un cuvant scurt, o parte dintr-un cuvant sau un grup de caractere.

Ca aproximatie, 1.000 de tokeni reprezinta circa 750 de cuvinte in limba engleza. In limba romana, aceasta corespondenta poate fi diferita.

Sa luam exemplul modelului GPT-5.4 mini, oferit de OpenAI. La tarifele folosite in exemplul nostru, in noiembrie 2026:

- 1.000 de tokeni transmisi catre AI costa 0,00075 dolari.
- 1.000 de tokeni generati de AI costa 0,00450 dolari.

Daca o aplicatie transmite 1.000 de tokeni si primeste un raspuns de 1.000 de tokeni, costul total este de aproximativ 0,00525 dolari, adica putin peste jumatate de cent american.

In acest exemplu, tokenii generati de AI costa de sase ori mai mult decat cei transmisi catre acesta.

Preturile depind de furnizor si de modelul AI utilizat. Pot exista si costuri suplimentare pentru anumite servicii.

Observam ca o singura prelucrare poate avea un cost foarte redus. Totusi, intr-o aplicatie utilizata frecvent, consumul se acumuleaza pe intreaga perioada de exploatare.

Cum putem optimiza consumul?

Un element important il reprezinta dimensiunea contextului transmis catre AI.

Sa presupunem ca un sistem informatic este organizat in zece module.

Daca AI-ul trebuie sa lucreze numai asupra modulului 1, nu este necesar sa primeasca documentatia completa a tuturor celor zece module.

Aplicatia poate transmite contextul general indispensabil, documentatia modulului 1 si numai informatiile din celelalte module necesare pentru intelegerea dependentelor.

Selectarea contextului se poate realiza si dinamic, cu participarea AI-ului.

De exemplu, la apelare, aplicatia ii poate transmite urmatoarea regula de lucru:

„Pentru rezolvarea sarcinii, solicita numai specificatiile modulului asupra caruia lucrezi. Nu solicita documentatia intregului proiect decat daca este necesara.”

AI-ul poate solicita apoi aplicatiei, prin functiile puse la dispozitie, specificatiile modulului respectiv. Aplicatia le identifica si i le transmite.

In acest fel, AI-ul primeste informatiile necesare fara ca intreaga documentatie a sistemului sa fie transmisa la fiecare apelare.

Aceasta metoda poate reduce consumul de tokeni si costurile de exploatare.

Unii furnizori ofera tarife reduse pentru informatiile reutilizate prin mecanisme de memorare temporara a contextului. In exemplul nostru, tokenii astfel reutilizati pot costa de zece ori mai putin decat tokenii obisnuiti de intrare. Pentru a beneficia de aceasta facilitate, este util ca partea neschimbata a contextului sa fie transmisa in aceeasi forma, la inceputul solicitarii.

12. Cine raspunde de functionarea sistemului?

AI-ul poate analiza informatii, genera documente, propune modificari si solicita executarea unor functii puse la dispozitie de aplicatie.

Dar aplicatia trebuie sa controleze ce operatii sunt permise, ce informatii pot fi accesate si cum sunt utilizate rezultatele.

Pentru operatiile importante, mai ales cele care modifica date sau codul sursa al aplicatiei, trebuie prevazuta posibilitatea verificarii si aprobarii de catre un operator uman.

Inca din etapa de analiza si proiectare trebuie stabilite regulile pentru verificarea rezultatelor, tratarea erorilor, protejarea informatiilor confidentiale si evidenta solicitarilor efectuate.

Integrarea AI nu elimina responsabilitatea proiectantului si a utilizatorului sistemului.

Concluzie

Integrarea inteligentei artificiale intr-o aplicatie informatica nu inseamna simpla deschidere a unei conversatii cu ChatGPT, Gemini sau Claude.

Ea presupune construirea unui mecanism prin care aplicatia transmite solicitari, pune la dispozitie informatii si functii, primeste rezultatele si le foloseste in continuarea propriilor prelucrari.

Fiecare apel AI reprezinta o operatie distincta. Continuitatea activitatii trebuie organizata, iar contextul trebuie gestionat cu atentie.

Utilizarea AI este justificata atunci cand aduce o capacitate utila sistemului informatic, nu doar pentru ca tehnologia este disponibila.

Aplicatia organizeaza si controleaza activitatea. Inteligenta artificiala realizeaza prelucrarile pentru care este solicitata.

Daca mecanismul de integrare este prevazut inca din faza de analiza si proiectare, implementarea sa poate fi realizata tot cu ajutorul inteligentei artificiale, folosita de aceasta data in rolul de programator.

Daniel Pop 15.11.2026

Analist de sistem