

Exposé Bachelorarbeit Jan Pfeifer

Ausgangssituation

Unternehmen, die SAP ECC Systeme nach SAP S/4HANA migrieren oder einfach nur Ihr System optimieren wollen, benötigen eine umfassende Analyse ihrer bestehenden Systemlandschaft.

Dabei unterstützt der Enterprise Analyzer (EA):

Extraktion von Systemdaten aus Kundensystem → Upload im EA → Speicherung in Tabellen → Aufbereitung über OData/CDS → Darstellung über Fiori/UI5-Apps.

Das Analysieren der Daten selbst ist jedoch größtenteils manuell und das führt dazu, dass Nutzer, größtenteils unabhängig von ihrer Vorerfahrung mit der Analyse von SAP Systemen, wissen müssen, wonach Sie suchen und was relevant ist, um den EA optimal zu verwenden.

Hypothese:

Ein KI-gestütztes Chat-Interface könnte diese Analyse vereinfachen, indem Nutzer Informationen natürlichsprachlich abfragen können und Hilfestellung bei dem Finden und der Interpretation der Daten bekommen.

Forschungsfrage

Wie kann die KI den Nutzer bei der Analyse von SAP Systemen unterstützen?

Unterstützen bezieht sich hier direkt auf direkte Wiedergabe von Informationen der Apps („Wie viele Einträge hat Tabelle X?“) sowie Interpretation („Was bedeutet das für das System, dass Tabelle X, Y Einträge hat?“) und indirekte Unterstützung („Auf welche Tabellen/Module sollte ich einen Blick werfen?“)

Methodik

1. Systematische Recherche & Technologie-Analyse (*explorativ*)

Ziel: Identifikation geeigneter KI- und SAP-naher Technologien.

Inhalt:

- Bewertung von Tools im SAP-Kontext
- Recherche vergleichbarer wissenschaftlicher Ansätze (State of the Art)

Ergebnis:

- Begründete Auswahl eines Technologie-Stacks
- Identifikation von Anforderungen und Use Cases

2. Prototypentwicklung (*konstruktive Forschung / Prototyping*)

Ziel: Umsetzung eines funktionalen Chatbot-Prototyps im Enterprise Analyzer.

Schwerpunkte:

- Anbindung an bestehende OData-Service
- Natural-Language-to-Query Mapping
- Dynamische Ausgabeformate (Text / Tabellen / ggf. Charts)

Es wird geprüft mit welchen Prompts und Aufgaben der Chatbot am besten umgehen kann.

3. Empirische Evaluation im Anwendungskontext (*qualitativ + quantitativ*)

Ziel: Überprüfung, in welchen Fällen der Chatbot den Nutzer bei bestimmten Apps unterstützen kann.

Methoden:

- Nutzerstudien / Stakeholder-Interviews
- Usertests

Ergebnis:

- Ableitung von Bewertungskriterien für Chatbot-Einsatz
- Identifikation von Grenzen & Verbesserungspotenzial

State of the art

Inhouse SAP AI Lösungen mit welchen man ein Model Trainieren und nahtlos in das oData Backend und UI5 Frontend integrieren kann.

[Generative AI | SAP BTP AI Best Practices](#)

Der EA ist für Externe nicht ohne weiteres Zugänglich.

Gliederung

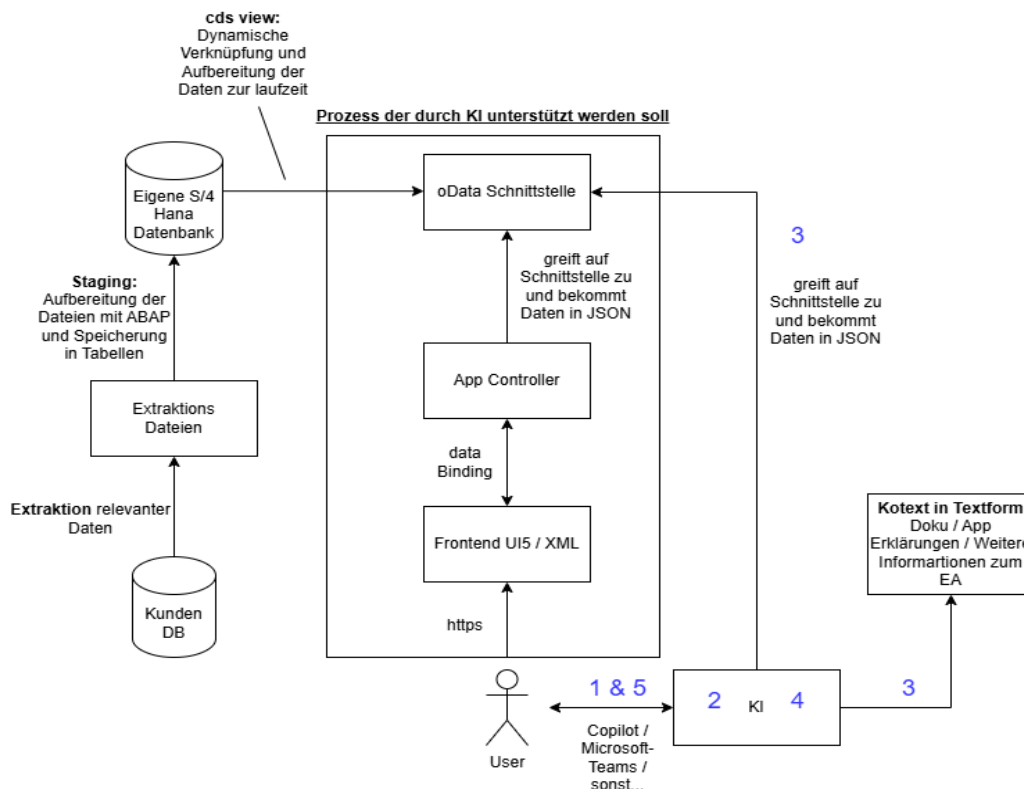
1. Abstract
2. Einleitung & praktische Relevanz
3. Grundlagen
 - a. SAP Fiori, UI5, Odata
 - b. Enterprise Analyzer
 - c. Grundlagen KI & LLMs
4. Methodik
5. Use Cases & Anforderungen
 - a. Stakeholderinterviews & User-Stories
 - b. Konzept Workflow / Einbindung in das User interface
6. Vision
7. Technologiewahl & Architektur

8. Prototypische Umsetzung
 - a. Datenfluss (Rohdaten → CDS → Chatbot)
 - b. Implementierung Chat-Interface
 - c. Connector zum OData Backend
 - d. Training Bot / System Prompt(s)
9. Evaluation & Ergebnisse
 - a. Vergleich zu bestehenden Apps
 - b. Darstellungsschwächen & Limitationen
10. Kritische Betrachtung & Grenzen
11. Ausblick (höhere Level: Cross-App-Analyse durch KI / Cross-Analysen Analysen durch KI)
12. Fazit

Zeitplan

Phase	Zeitraum
Recherche + Anforderungsanalyse	Woche 1-2
Architekturdesign + Technologieauswahl	Woche 3-4
Prototypentwicklung	Woche 5-9
Evaluation & Tests	Woche 10-11
Überarbeiten / Finalisierung Bachelorarbeit	Woche 12

Prozess der Daten im EA



in blau symbolisieren die im Kapitel "Einsatz von KI" beschriebenen Prozesse

Hier ist der gesamte Prozess der Daten von Kundensystem bis EA-Frontend zu sehen. In der Bachelorarbeit liegt darin, herauszufinden, inwiefern KI den Nutzer unterstützen kann, bestimmte Informationen über das Kundensystem zu erhalten, direkt durch Wiedergabe der Daten oder Hilfestellung bei der Benutzung des EAs.

Abbildung 1: Der Prozess der Daten im EA. Die Zahlen

Einsatz von KI

Hier wird der Prozess von Nutzerinput bis KI-Output dargestellt (bereits mit einigen Hürden/potentiellen Problemen), bewertet und mit Beispielen anschließend anschaulich gemacht.

1. Nutzer Klartext eingabe im Frontend

Kann der Nutzer sein Problem gut darstellen?

2. Anfrage wird von KI verarbeitet / interpretiert

Versteht die KI was der Nutzer will?

3. Query generierung an das Backend / „lookup“ im gegebenen Kontext (Doku)

Kann die KI basierend auf dem Nutzerinput die korrekten Daten abrufen?

4. Verarbeitung / Interpretation der Daten

Wie bereitet die KI die Rückgaben auf?

5. Wiedergabe der Daten als Text / Tabelle(n) (/ Chart(s)?)

Wie gibt die KI die Daten aus? Versteht der Nutzer die Darstellungen?

Risikoabschätzung

Bei der Nutzung von KI entstehen einige Unschärfen und Risikofaktoren. Es fängt bereits bei dem Nutzerinput an. Hier kann man als Entwickler nie wissen, erst recht bei Freitext-Eingaben für einen KI-Prompt, ob der Nutzer sich so ausdrückt, wie man als Entwickler erwartet. Daraufhin wird die Anfrage mit dem System-Prompt verarbeitet und interpretiert.

Im Falle von unserem Use-Case soll die KI hier von uns alles Benötigte an Kontext (mit Kontext ist hier gemeint, dass der KI die Backend-Schnittstelle(n) und deren Bedeutung erklärt werden und ebenfalls, wie sie auf diese zugreifen kann), um zu wissen, auf welche Daten der Nutzer hinauswill und ihm im Zweifel eine Empfehlung geben zu können, wo er diese Informationen im EA finden könnte. Wichtig ist hier zu beachten, dass die KI selbstverständlich (noch) nicht über die Grenzen des EAs hinweg helfen kann und auf die Daten beschränkt ist, die durch das Backend zur Verfügung gestellt werden sowie den Kontext, der in Form unserer Dokumentation und ggf. speziell angefertigten Texten vorliegt.

Basierend auf diesem Kontext erstellt die KI eine Backend-Query (Kapitel „Beispiele“ 1-3) oder kann direkt eine Antwort ableiten (Kapitel „Beispiele“ 4). Hier besteht die klassische Gefahr, dass die KI syntaktisch oder inhaltlich falsche Queries erstellt. Dem muss man klassisch mit Promptengineering (Grundsyntax vorgeben, Beispiele im Systemprompt aufnehmen) begegnen.

Sollte nun die KI die richtige Query gesendet haben und dementsprechend eine Response bekommen haben, gilt es nun, die Daten, dem Prompt entsprechend, zu interpretieren. Wie auch bei der Verarbeitung vom Userinput besteht hier die Gefahr von Halluzinationen und Fehlinterpretation. Jedoch nicht in dem Ausmaß wie bei der Interpretation des user-inputs, da man, wenn man schon erfolgreich bei diesem Schritt angekommen ist, davon ausgehen kann,

dass der Hauptteil des „Denkens“ vor der Query bereits erledigt wurde und die KI nur noch die Daten in entsprechendes Format bringen muss.

Hier besteht nun die Aufgabe darin, dass die KI die Daten dem Userprompt entsprechend formatieren muss und falls nicht explizit erwähnt wurde, eine sinnvolle Möglichkeit wählen soll.

Fazit:

- Prompt-Engineering wird ein großes Thema.
- Spezielle Schnittstellen im Backend erstellen, deren Felder/Daten speziell für den Zugriff der KI gebaut sind.
- Nutzer befragen zu möglichen Prompts, die eingegeben werden.
- Nutzer und Entwickler befragen zur Qualität der Antworten.

Beispiele

Die Beispiele sind mögliche User-Inputs, auf die die KI eine zufriedenstellende Antwort (Inhalt und Format) liefern soll. Die Möglichkeiten der KI-Antworten (Text, Charts, Dateien, ...?) gilt es zu überprüfen.

1. Direkte Datenabfrage aus dem Backend

Frage: „Wie viele Tabellen aus dem Namespace /OCST/ haben mehr als 2000 Einträge?“

Antwort: „Es gibt 234 Tabellen mit dem Namespace /OCST/ und >2000 Einträgen.“

Die KI baut eine OData-Query an die Backend-Schnittstelle, von der sie direkt aus der Antwort die Informationen wiedergibt. (Beispiel unter 2.)

2. Abfrage aus Backend mit bitte um Formatierung (Tabelle)

Frage: „Welche Tabellen aus dem Namespace /OCST/ haben zwischen 2000 und 10 000 Einträge? Gib mir die Tabellen mit allen Feldern und sortiere absteigend nach Einträgen.“

Beispiel Query: TableStatisticsRecordSet?sap-client=100&\$top=10&\$orderby=NumberOfRows desc&\$filter=(NumberOfRows ge 2000m and NumberOfRows le 10000m) and Namespace eq '/OCST/'HTTP/1.1

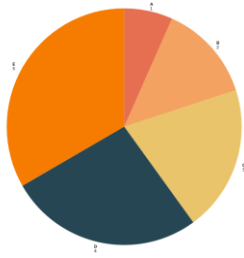
Antwort: „Hier ist eine Tabelle aller Tabellen aus dem Namespace /OCST/ welche zwischen 2000 und 10000 Einträgen haben, in absteigender Reihenfolge für das Feld *Records*“

Name	Records	Namespace	Delivery Class	Module
ERSTVC	8745	/OCST/	A	CO
ERSTVCESD	4664	/OCST/	A	CO
FRTSD	3233	/OCST/	E	CO
VFESD	2211	/OCST/	W	ACC

3. Abfrage aus Backend mit bitte um Formatierung (Chart)

Frage: „Was ist der Anteil der Delivery Classes von allen Tabellen, dargestellt in einem pie-chart.“

Antwort: „Hier ein Chart der die %-Anteile der verschiedenen Delivery Classes über alle Tabellen anzeigt“



4. Allgemeine Frage zum EA (was machen bestimmte funktionen?)

Frage: „Was bedeutet / ist der Inhalt der Spalte „BO Type“ der Business Object Overview App?

Antwort: „Laut der Dokumentation gibt hier 3 verschiedene Möglichkeiten:

- **Customizing Data:** For example, standard texts for routing.
- **Master Data:** For example, material master.
- **Transactional Data:** For example, purchase order.

Frage: „Welche Informationen liefert die Business Object App?“

Antwort: „ Diese App wird verwendet, um zu ermitteln, welche Business Objects (BOs) im System genutzt werden. Sie liefert Informationen [...]. Zudem können Sie sowohl die Anzahl der Business Objects pro Modul als auch die Anzahl der Datensätze pro Modul einsehen.“

Frage: „Ich will den Aspekt XY des Systems überprüfen, welche App ist dafür geeignet?“

Antwort: „Um Aspekt XY des Systems zu überprüfen ist App ZZ geeignet da hier *{Begründung}*.“

5. Bitte um Interpretation der Daten (woher hat die KI die Basis)

Frage: „In meinem System gibt es ungewöhnlich viele Tabellen mit der Delivery Class A. Was könnte das bedeuten?“

Antwort: ???

Diese Frage benötigt eine Interpretation der Daten und zusätzliches Training der KI neben dem Übergeben von Kontext. Mehr dazu in „Qualitätssicherung“.

6. Unmögliche Fragen

Frage: „Wieviel Umsatz hat der Kunde im Jahr gemacht?“ (Allgemein Fragen zu Daten die die KI nicht hat)

Qualitätssicherung

Generell soll die KI immer eine Quelle nennen und/oder wiedergeben, wie sie zu der Antwort kam. Zudem kann man noch zu verschiedenen Fragentypen weitere Maßnahmen ergreifen, um die Qualität der Antworten zu messen:

1. Fragen mit genau einer richtigen Lösung

Ob die KI Anfragen im Format von Punkten 1–3 im Kapitel *Beispiele* richtig bearbeitet, ist einfach zu prüfen.

Man kann einfach mehrere Testanfragen schreiben, mit verschiedenen Wordings, und die von der KI zurückgegebenen Daten direkt mit denen aus dem Backend vergleichen, um festzustellen, ob unterwegs etwas schiefgelaufen ist.

2. Fragen mit offenen Antworten

Bei Fragen im Format von Punkt 4 und 5 wird die Bestimmung der Qualität aufgrund dessen schwer, da es keine konkrete korrekte Antwort gibt.

Der Sinn der Antwort muss evaluiert werden.

Das ist recht einfach bei Fragen, die man sich z.B. direkt aus der Doku herleiten kann „Was kann ich mit der Business Object Overview App machen?“

Schwerer bei Fragen, die eine Empfehlung erwarten: „Ich will den Aspekt XY des Systems überprüfen, welche App ist dafür geeignet?“, da sich hier die KI basierend auf all ihren Informationen selbst eine Antwort zusammensetzen muss und nicht einfach z. B. direkt aus dem Doku-Kontext eine App-Beschreibung holen kann wie bei der Frage: „Was kann ich mit der Business Object Overview App machen?“

Und mit Abstand am schwersten sind Fragen unter Punkt 5 wie: „In meinem System gibt es ungewöhnlich viele Tabellen mit der Delivery Class A. Was könnte das bedeuten?“, bei denen eine Analyse der KI erwartet wird. Dafür braucht man i. d. R. tiefgehendes Wissen. Man müsste die KI mit vielen Daten und Auswertungen trainieren, um zuverlässig korrekte Ergebnisse zu liefern. Der Vollständigkeit halber wurde dieser Punkt als Beispiel aufgenommen. Diese Funktionalität wäre die langfristige Maximalausprägung dieses Prozesses.

Der Ansatz zur Qualitätssicherung ist, einem oder mehreren erfahrenen Nutzern des EA's einen Katalog an Fragen zu stellen. Die Antworten werden dann mit den Antworten der KI auf dieselben Fragen verglichen, und es wird geprüft, wie sehr sich der Inhalt (Erklärungen/Empfehlungen) ähnelt und ob die KI unzureichende oder eventuell sogar bessere Antworten liefert.

3. Fragen ohne Antwort

Zu manchen Fragen darf es keine Antwort geben: „Wieviel Umsatz hat das Unternehmen dieses Jahr gemacht?“. Solche und andere privaten Daten (Mitarbeiter, Finanzen, Strategiepläne ...) werden nicht in der Extraktion (Abb. 1) berücksichtigt, und es sollen Missverständnisse durch mögliche Halluzinationen der KI vermieden werden.

Zur Qualitätssicherung wird ein Katalog an Fragen erstellt, auf die die KI keine Antwort geben soll. Die Fragen werden auch verschieden formuliert. Die KI soll deutlich machen, dass sie zu diesen Themen keine Informationen hat.