

WI-Studierendenforum

Ausgabe 76

Kontaktinformationen

Zuschriften zum Studierendenforum
bitte an:
Universität Karlsruhe (TH),
Institut für Informationswirtschaft
und -management (IISM),
Prof. Dr. Ch. Weinhardt, Redaktion
Studierendenforum,
z. Hd. Dr. Henner Gimpel,
Englerstraße 14, 76131 Karlsruhe,
07 21 6 08 83 74
gimpel@iism.uni-karlsruhe.de

In dem folgenden Beitrag berichten die Autoren – Christoph Riedl, Johannes Schöfmann, Matthias Müller – über eine Projektpraktikum, das Sie im Rahmen ihres Studiums der Wirtschaftsinformatik an der TU München durchgeführt haben. Inhaltlich geht es dabei um die Integration eines Avatars, d. h. eines graphischen Stellvertreters einer natürlichen Person, in einen SAP-R/3-basierten Onlineshop. Darüber hinaus gehen die Autoren auf die Bedeutung des Projektpraktikums für ihr Studium sowie Chancen, Probleme, Herausforderungen der Teamarbeit im Studium ein.

■ Integration eines Avatars in ein SAP-R/3-System – Projektpraktikum

■ 1 Einleitung

Zentrales Ziel des Projekts war es, zwei bisher weitgehend getrennte Welten zusammen zu führen: Komplexe ERP-Systeme und Chatbots. Diese Integration wurde am Beispiel des ERP-Systems SAP R/3 und des Open Source Chatbots A.L.I.C.E. durchgeführt. Dadurch sollte Besuchern eines Onlineshops eine natürlichsprachliche Beratung während des Kaufprozesses angeboten werden. Mit dem Einsatz von Chatbots kann eine verbesserte Service- und Beratungsqualität sowie eine Entlastung der Mitarbeiter erreicht werden.

Das SAP-R/3-System stellt die Datenbasis dar, um den Verkaufprozess abbilden zu können. Diese enthält die betriebswirtschaftlichen Daten, wie Produkte, Lieferungen, Zahlungsarten usw., die sich der Chatbot für den Beratungsprozess benötigt. Gleichzeitig besitzt der Chatbot auch eine Schnittstelle zum Webshop, die es ermöglicht, die Navigation des Kunden zu beeinflussen. Es kann somit beispielsweise eine Weiterleitung auf die Detailansicht eines Produktes realisiert werden.

Folgende Technologien kamen bei dem Projekt zum Einsatz:

- SAP R/3 Enterprise als ERP-System
- Zugriff auf SAP aus Java mittels SAP Java Connector (JCo)
- SAP Internet Transaction Server (ITS) mit Webshop Internet Application Component (IAC) als Internetfrontend und
- die Java-Implementierung des A.L.I.C.E. Chatbotsystems ProgramD [ALICE].

Die Vielzahl an eingesetzten Technologien und die Vielzahl der Schnittstellen machte die Umsetzung sehr schwierig, da Lö-

sungsmöglichkeiten nicht nur zwischen den Teams abgestimmt, sondern erst erarbeitet werden mussten. Daraus ergaben sich viele Abhängigkeiten zwischen den Teams und Verzögerungen des gesamten Ablaufs.

Das Projekt wurde im Rahmen eines Programmierpraktikums im Sommersemester 2006 am Lehrstuhl für Wirtschaftsinformatik der TU München durchgeführt und hatte einen Umfang von 6 SWS und 12 ECTS. Das Praktikum wurde von 16 Studierenden aus Master- und Diplomstudiengängen der Bereichen Betriebswirtschaft, Wirtschaftsinformatik und Informatik im 8. und 9. Fachsemester durchgeführt. Dabei wurden fünf Teams mit den Themen Avatar, SAP-Webshop, BAPI¹-Programmierung, Chatbot sowie der Schnittstelle zum SAP-R/3-System gebildet. Für die Bearbeitung stand das gesamte Sommersemester 2006 zur Verfügung.

Chatbots, oder kurz Bots sind – meist textbasierte – Dialogsysteme im Internet. „Chatbots antworten auf natürlichsprachliche Sätze und stellen selbst Fragen. Dies geschieht mithilfe einer umfangreichen Wissensbasis, in der mögliche Fragen und dazupassende Antworten gespeichert werden“ [Pixe06]. Um einem Chatbot ein menschliches Auftreten zu geben, werden diese oft durch individuelles Aussehen und persönliche Eigenschaften als Avatar realisiert. Avatare werden so zu einer virtuellen Person und bei guter Umsetzung tatsächlich als menschlich empfunden.

Zur Repräsentation der Wissensbasis für die benötigten Dialoge wurde Artificial Intelligence Markup Language (AIML) verwendet. AIML ist eine XML basierte Spra-

che, die für den Einsatz in A.L.I.C.E. basierten Chatbots entwickelt wurde (alicebot.org).

Die Wissensbasis lässt sich mit der AIML-Sprache, über *categories* (Tags) aufbauen. Diese setzen sich wiederum aus einem *pattern* (Stimulus) und einem *template* (Antwort) zusammen (siehe Bild 1).

Nach dieser kurzen Einführung in AIML stellt der nächste Abschnitt die Projektschwerpunkte ausführlich dar. Diesem Teil folgt eine Zusammenfassung.

2 Ausarbeitung wichtiger Projektabschnitte

Das Projekt besteht aus drei Projektschwerpunkten: Die Geschäftsprozesse innerhalb des Avatars, den benötigten Schnittstellen zwischen den einzelnen Systemkomponenten und der Gestaltung der graphischen Oberfläche. Diese werden im Folgenden kurz vorgestellt.

2.1 Geschäftsprozess innerhalb des Avatars

Der Avatar unterstützt den Verkaufsprozess durch eine kompetente Beratung des Kunden. Dazu ist er mit einer umfangreichen Wissensbasis (über 800 Regeln) ausgestattet, die genaue Informationen über die angebotenen Produktkategorien enthält (nicht jedoch über spezielle Produkte innerhalb dieser Kategorien). Bild 2 gibt einen Überblick über die von uns gewählten Kategorien.

In der Kategorie „Workstation“ wird beispielsweise zwischen einfachen Produkten für normale Officeanforderungen und leistungsfähigeren Produkten für die Graphikbearbeitung unterschieden. So wird der Kunde zum Beispiel gefragt „Mit welchen Programmen arbeiten sie hauptsächlich?“. Das Avatarsystem erkennt dabei Schlüsselwörter wie „Word“, „Excel“, „Microsoft“, „Internet“ oder „E-Mail“ und präsentiert dem Kunden eine Auswahl der Produkte aus der „Desktop“ Kategorie. Antwortet der Kunde hingegen mit Schlüsselwörtern wie „Photoshop“, „CAD“ oder dergleichen, werden ihm Graphikworkstations angeboten.

Der zweite Projektschwerpunkt war die Gestaltung der Schnittstellen zwischen den Komponenten des Gesamtsystems. Die Probleme und Lösungen werden im folgenden Absatz vorgestellt.

```
<category>
  <pattern>HALLO</pattern>
  <template>Hallo, ich bin Alice. Wie heisst du?
</template>
</category>
```

Bild 1 Codebeispiel AIML-Regeln

2.2 Schnittstellen zwischen den Komponenten

Der Avatar bietet drei Funktionen, die alle über Schnittstellen durch die AIML-Regelbasis gesteuert werden: Steuerung der Avatar-Animationen, Laden von neuen Seiten im Web-Frontend und den direkten Zugriff auf Informationen aus dem SAP-R/3-System. Diese Schnittstellen werden in den folgenden Kapiteln genauer erklärt.

2.2.1 Steuerung der Avatar-Animationen

Die Steuerung des Avatars erfolgt durch JavaScript, das in die Ausgabe der AIML-Regeln integriert und dann von einer Funktion im HTML-Dokument ausgelesen wird. Da AIML-Ausgaben beliebigen HTML-Code enthalten dürfen – also auch -Anweisungen – ist dies ohne Probleme möglich. Der in die HTML Seite des Shops eingebettete Avatar wird dabei über genau eine JavaScript Variable gesteuert, die im Output der AIML-Regeln gesetzt wird.

Im nächsten Schritt wird dann die im AIML-Output gesetzte Variable von einer JavaScript Funktion, die beim Laden der Seite OnLoad ausgeführt wird, ausgelesen und evaluiert. Ist in der AIML-Ausgabe kein Wert für eine Animation gesetzt, weil für diese Situation keine spezielle Animation vorhanden ist, wird eine Standardanimation geladen.

2.2.2 Navigation des Web-Frontends durch den Avatar

Ziel war es, eine Technik zu entwickeln, mit welcher der Webshop durch den Avatar gesteuert werden kann. Zum einen, weil es den Komfort für den Benutzer erhöht, wenn er automatisch auf die gewünschten Seiten weitergeleitet werden kann. Zum anderen müssen dadurch nicht ausführliche Produktdetails oder Produktübersichten in die Avatarantworten integriert werden, sondern können übersichtlicher im Hauptbereich des Browsers dargestellt werden, indem einfach die entsprechenden Seiten geladen werden.

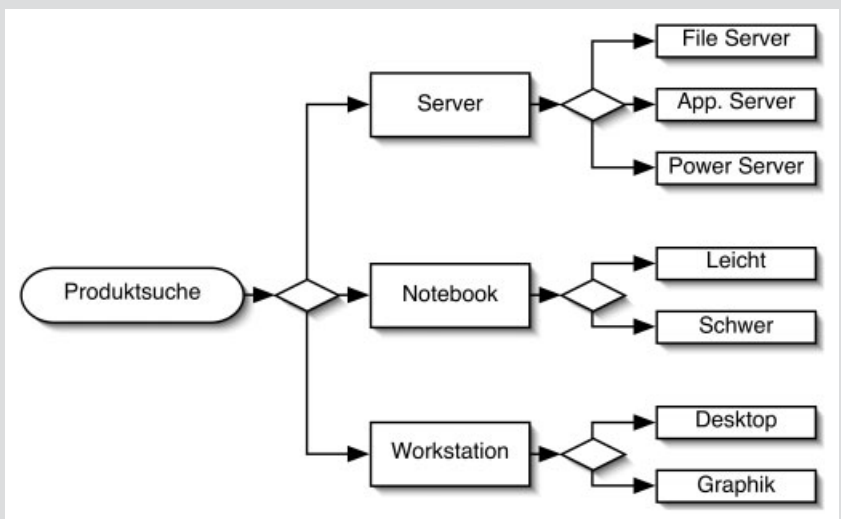


Bild 2 Verfügbare Produktkategorien

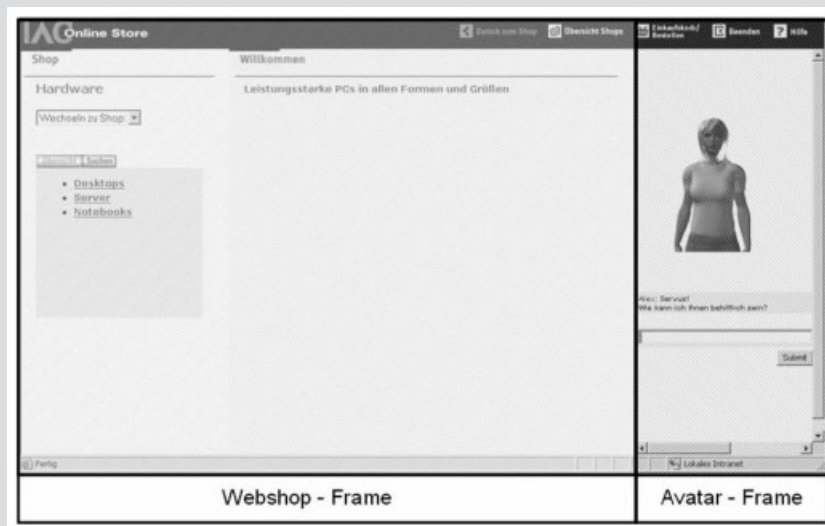


Bild 3 Einbindung des Avatars mittels Frameset

Da das HTTP-Protokoll keine Push-Möglichkeit vorsieht, muss die Weiterleitung clientseitig ausgelöst werden. Dazu wurde – wie auch bei der Steuerung der Avatar-Animationen – JavaScript verwendet. Die Antwort des Avatars enthält deshalb JavaScript-Code, der sobald die Webseite geladen und im Browser ausgeführt wird, die Weiterleitung auslöst. Das Laden der neuen Seite wird durch eine Zuweisung zum location-Property ausgelöst: `window.location = neue Adresse`

2.2.3 SAP-R/3-Integration

Um den Beratungsprozess angemessen zu unterstützen, benötigt der Avatar die Möglichkeit, auf Daten des SAP-R/3-Systems zuzugreifen und diese in seine Antworten einzubinden. Um diese Zugriffe zu realisieren, wurde eine Java-Schnittstelle entworfen, die über XML-Tags in die AIML-Regeln integriert werden kann. Die in den AIML-Regeln verwendeten Tags werden von einer modifizierten Version des ProgramD geparsed. Die Schnittstelle generiert aus der vom SAP-System gelieferten Rückgabe eine natürlichsprachliche Antwort, die über den Avatar ausgegeben wird. Auf Seite des SAP-R/3-Systems sind diese Erweiterungen als remotefähige Funktionsbausteine realisiert, die aus dem ProgramD über eine Java JCo Schnittstelle aufgerufen werden.

Wir entschieden uns für eine generische Schnittstelle, die eine freie Konfiguration des gewünschten Funktionsaufrufs, der übergebenen Parameter und der gewünsch-

ten Rückgabewerte erlaubt. Dies gewährt uns ein Maximum an Flexibilität und neue BAPI Aufrufe können somit sehr schnell implementiert werden.

Ein am Markt angebotenes Produkt der Firma Novomind bietet ebenfalls die Möglichkeit, einen Avatar mit einem SAP-R/3-System zu verbinden. Eine ausführliche Evaluierung dieses Produkts fand nicht statt.

2.3 Graphische Gestaltung des Avatars

Ein wichtiger Bestandteil der zu erstellen Lösung war die visuelle Repräsentation des Chatbots, also der Avatar. Der Avatar sollte so entworfen werden, dass sich Benutzer gerne mit ihm unterhalten und ihm relevante Fragen stellen. Deshalb wurde bei der Erstellung der visuellen Repräsentation auf ein angenehmes und freundliches Erscheinungsbild Wert gelegt. Bild 3 zeigt unseren Avatar, wie er in den Webshop eingebunden wurde.

Gleichzeitig wurde durch die Erstellung einer Vielzahl von Animationen, dem Avatar eine eigene Persönlichkeit gegeben, um so die Benutzer weiter zur Interaktion zu animieren.

Zu den erstellten Animationen zählen: Begrüßung, Freude, Zeigen auf Artikel im Webshop, Zustimmung, Ablehnung, Verabschiedung usw. Diese Animationen wurden mit einem 3D-Programm modelliert, gerendert und schließlich als animierte GIF-Dateien exportiert. Besonderer Wert

wurde hierbei auf eine detailgetreue Mimik und Gestik gelegt, um den Avatar möglichst realistisch und teilweise lustig auf Benutzereingaben reagieren zu lassen. Dadurch sollte erreicht werden, dass Benutzer gerne mit dem Avatar interagieren und der Beratungsprozess optimal genutzt wird.

3 Ergebnisbetrachtung

In diesem Kapitel wird dargelegt, welche Probleme bei der Umsetzung auftraten und wie diese gelöst wurden. Dabei wird auch begründet, weshalb die jeweilige Lösung gewählt wurde.

3.1 Problem der Prozessgestaltung

Das erste Problem bestand in der Entscheidung, welche Produkte im umzusetzenden Onlineshop angeboten werden sollten. Diese durften nicht zu komplex sein, da dies sonst die Fähigkeiten einer avatar-gestützten Produktberatung übersteigen würde. Andererseits sollten die Produkte auch nicht zu einfach sein, da sonst keinerlei Beratung notwendig gewesen wäre.

Wir entschieden uns daher, Produkte aus drei Kategorien anzubieten: Notebooks, Desktops und Server. Diese Produkte haben den Vorteil, dass sie alle eindeutige und dennoch unterschiedliche Attribute haben, anhand derer sie klassifiziert werden können (z. B. "leicht" und "schwer" bei Notebooks).

3.2 Probleme der technischen Umsetzung

Das mit Abstand größte Problem bestand in der Integration aller Komponenten des Gesamtsystems. Hierzu mussten nicht nur die Schnittstellen zwischen den Modulen (Webshop, Avatar, Java-Backend Avatar-Connect und SAP R/3 Backend) definiert und implementiert werden, sondern zuerst technische Lösungsmöglichkeiten erarbeitet werden, da nicht bekannt war, wie diese Probleme generell gelöst werden können.

Für den Zugriff auf das SAP-Backend wurde in mehreren Treffen des AIML-Teams (drei Teammitglieder) mit dem Java-Team (fünf Teammitglieder) die Lösung durch den Einsatz von XML erarbeitet. Diese Idee wurde in der tatsächlichen Implementierung schrittweise zu der unter 2.2.3 beschriebenen Schnittstelle weiterent-

wickelt. Im Bereich der Vorwärtsnavigation wurde zur Lösung ein Prototyp aus statischen HTML-Seiten programmiert, in dem das Konzept der unter 2.2.2 beschriebenen Schnittstelle entwickelt wurde. In diesem Prototyp wurde die Aktivierung von Links über JavaScript evaluiert und die nötigen JavaScript Klassen programmiert. Es wurde außerdem das Konzept entwickelt, wie aktuelle Ortsangaben – insbesondere die Session-ID des Webshops – direkt aus dem dynamisch generierten HTML-Code ausgelesen werden können. Die aus dem Prototyp resultierenden JavaScript-Klassen konnten dann mit geringen Anpassungen in das Gesamtsystem übernommen werden.

Die Steuerung der Avatar-Animationen stellte hingegen kein technisches Problem dar, da hierfür eine im Rahmen einer früheren Hausaufgabe des Praktikums entwickelte Lösung unverändert übernommen werden konnte.

3.3 Probleme bei der Frontend-Integration

Ein weiteres Problem, das es zu lösen galt, war die Integration des Avatars in den ITS-Webshop. Die erste Erkenntnis hierbei war, dass Avatar und Webshop unabhängig voneinander geladen werden müssen. Das Abschicken eines Dialogs an den Avatar soll nicht zum Nachladen des gesamten Shops führen. Umgekehrt darf beim Klick auf einen Link im Onlineshop nicht der Avatar neu geladen werden, da hierdurch der Dialogkontext verloren gehen würde. Dies erforderte somit den Einsatz von Frames. Auch hier wurde ein Prototyp geschrieben und in einem evolutionären Ansatz die endgültige Lösung entwickelt.

3.4 Schnittstellen und Abhängigkeiten zwischen den Teams

Eine weitere Herausforderung bei der Umsetzung war die Vielzahl der Schnittstellen, die abgestimmt werden mussten. Das AIML-Team war hiervon besonders stark betroffen, da die AIML-Regeln die inte-

grierende Komponente sind, die die Module Webshop, Avatar und Backend verbinden. Die durch die AIML-Regeln gesteuerte Produktberatung sollte schließlich alle Möglichkeiten der Vorwärtsnavigation ausnutzen und alle vom BAPI-Team entwickelten BAPIs aufrufen. Dadurch entstanden große Abhängigkeiten von den anderen Teams. Dies führte zu starken Verzögerungen, von denen das gesamte Projekt stark betroffen war.

Die auftretenden Probleme während der Projektarbeit waren hauptsächlich kommunikativer Art, was das Projektmanagement erheblich erschwerte. So wurden die Anforderungen nicht zentral bzw. überhaupt nicht erfasst und allgemein kommuniziert, sodass verschiedene Gruppen ihre Arbeit erst spät beginnen konnten.

4 Zusammenfassung und Ausblick

Im Rahmen des Projekts wurde die Integration eines Chatbots in einen SAP-Webshop zur Unterstützung und Beratung des Kunden während des Auswahl- und Kaufprozesses umgesetzt. Hierbei wurde der Chatbot über eine selbst entwickelte Schnittstelle direkt mit dem SAP-System verbunden, um den Zugriff auf dessen Datenbasis zu ermöglichen. Die Schnittstelle ist flexibel über XML zu konfigurieren, sodass sie für weitere Funktionen offen ist.

Die konkrete Umsetzung erfolgte in unserem Fall mit einem SAP-R/3-System als Backend und einem SAP ITS Webshop. Jedoch ist die von uns entwickelte Lösung so flexibel, dass jederzeit auch andere Systeme verwendet werden könnten.

Für die Frontend-Steuerung des Avatars ist es lediglich notwendig, den dynamisch generierten HTML-Code anzupassen. Je leichter dies möglich ist, desto einfacher ist die Integration. Die nötigen Anpassungen sollten bei jedem marktüblichen Webshop wie z. B. Openshop oder individuell entwickelten Shopsystemen (z. B. in PHP) ohne Probleme möglich sein.

Neben fachlichen Gesichtspunkten konnten Erfahrungen bezüglich der Durchführung von Projekten mit einem interdisziplinären Team gesammelt werden. Hierbei stach besonders hervor, wie wichtig die Abstimmung untereinander auf allen Ebenen ist. So sollten Anforderungen früh erfasst und kommuniziert werden. Ebenso ist eine frühe Definition der Schnittstellen zwischen den Teams notwendig, um eine reibungslose Projektarbeit zu ermöglichen.

Abschließend lässt sich sagen, dass das Projekt in einem spannenden Umfeld abgelaufen ist und erfolgreich abgeschlossen werden konnte: das System unterstützt die Anforderungen und bietet eine gute Basis für einen weiteren Ausbau.

Anmerkung

¹ BAPI: Business Application Programming Interface. BAPIs sind in ABAP programmierte SAP-Business-Objekte, die es externen Programmen erlauben, auf die Daten und Geschäftsprozesse eines SAP-R/3-Systems zuzugreifen.

Literatur

- [ALICE] A.L.I.C.E. Artificial Intelligence Foundation. <http://www.alicebot.org>, Abruf am 2007-01-24.
- [Pixe04] Pixelwings: Persönlichkeit: Pixelwings stellt den ersten virtuellen Mitarbeiter vor. <http://www.pixelwings.com/?content=aktuell/index.php?id=134>, 2004, Abruf am 2007-01-24.
- [Stra03] *Strake, Tim*: Die Qual der Wahl: Online-Beratungssysteme im Vergleich. <http://www.ecin.de/shops/beratungssysteme/>, 2003, Abruf am 2007-01-24.

Autoren

Christoph Riedl, Johannes Schöfmann,
Matthias Müller
{vorname.nachname@in.tum.de}

Zum Zeitpunkt des Projektpraktikums alle Studierende der Wirtschaftsinformatik (M.Sc.) an der Technischen Universität München.