

Der Kunde ist (doofer?) König

Ingrid Hayek
Universität Innsbruck
Innsbruck, Österreich

Schlüsselworte

Fachjargon, verbale Kommunikation, dynamische Hilfetexte

Einleitung

SQL, HTM und IOP

LOV, XML, ojemine

MAA, XSL und SAP,

DDL, RAC, oleole

ASM, CSO und SGA,

PGA, RDS und DBA

EDV DBMS, ihr könnt mich mal

MfG - mit freundlichen Grüßen...

(Die Fantastischen Vier - adaptierte Version von IH, 2012)

Dies ist ein Crash-Sprachkurs für die verbale Kommunikation zwischen Applikationsentwicklern und sogenannten unbeleckten Kunden. Bei der Beschreibung der Probleme und deren möglichen Lösungen sagen beide das Richtige und verstehen doch oft das Falsche. Eine wesentliche Voraussetzung für eine zufriedenstellende Applikationsentwicklung ist folgende: nicht der Kunde passt sich der Datenbank und der Entwicklungsumgebung an, sondern die Datenbank den Wünschen des Kunden. Es ist nicht wesentliche Voraussetzung oder Aufgabe des Kunden die Hintergründe der Datenbank zu verstehen, sehr wohl aber ist es essentiell, dass der Applikationsentwickler sich in das Fachgebiet des Kunden einarbeitet, um die Probleme zu verstehen und Lösungen anzubieten. Diese Lösungen müssen in der Sprache des Kunden angeboten werden.

Eine saubere Datenbankapplikation kann vom Kunden ohne besondere technische Vorkenntnisse bedient werden, ebenso wie wir alle gute Autofahrer sind (zumindest bilden wir uns das ein) und nur die wenigsten von uns sich noch an die rudimentären technischen Details aus Fahrschulzeiten erinnern (das geben wir oft sogar zu). Für Autofahrer gibt es (beinahe) unbegrenzte Freiheiten, wohin, wie viel und wie schnell sie fahren. Dabei werden ihnen nur minimale technische Kenntnisse abverlangt (Tanken, für Fortgeschrittene: Ölstandkontrolle, Scheibenwaschmittel nachfüllen). Von unseren Kunden sollten wir IT-mäßig auch nicht mehr verlangen.

Wie beginne ich ein "royales" Gespräch?

Manche Könige haben überhaupt keine Ahnung von Datenbanken und alles "mit Computer" abseits von Word, Facebook & Co. ist ihnen suspekt. Andere haben ein wenig mit Excel oder Access gearbeitet und versuchen womöglich ihre Vorstellungen über eine Datenbank-Applikation daran anzuknüpfen. In beiden Fällen, besonders aber im zweiten, sollte ich mich bemühen, das Gespräch völlig auf den Inhalt zu fokussieren und die Technik, zumindest verbal, auszuklammern. Je besser ich mich mit der Thematik des Königs auseinandersetze, desto besser kann ich die logischen Zusammenhänge und die prozeduralen Abläufe im Hinterkopf(!) auf die Datenbank ummünzen und desto leichter fällt es mir in der Sprache des Kunden zu kommunizieren.

Entity Relationship Diagramme sind eine wertvolle visuelle Unterstützung um Zusammenhänge anschaulich darzustellen. Es empfiehlt sich also, im Beisein des Königs einfache ER-Diagramme zu zeichnen, dabei aber *keinesfalls* von "ER-Diagrammen" zu sprechen. Vorgefertigte ER-Diagramme, womöglich gar noch komplizierte, verwirren und schrecken ab.

Beim Erstellen eines ER-Diagrammes lockern Vergleiche aus dem Alltagsleben das Gespräch auf und lassen Technisches nicht-technisch erscheinen:

- Unterschied Datenbank - Webformular (Datenbank = Mobiliar eines Hauses, Webapplikation = Fassade mit Türen und Fenstern)
- Tabellen sind wie ein Bücherregal. Wenn ich ein passendes anfertigen will, muss ich zuerst wissen, wie viele Bücher ich habe (oder haben werde) und nach welchen Kriterien ich sie ordnen will
- 1:n-Beziehung: Eine Mutter kann mehrere Kinder haben, ein Kind jedoch nur eine Mutter
- Master-Detail: Ein Kind kann erst getauft werden, wenn es auf der Welt ist
- Cascading Delete: ich darf nicht eine Wäscheleine abschneiden, an der noch Wäsche hängt

Welche Gefahren lauern vom Königshaus?

Vorsicht bei "Großwesiren" = scheinbar versierten Kunden, die ein bisschen Excel "können" und am liebsten selbst die Datenbank erstellen würden.

Vorsicht bei Ausdrücken wie "eigentlich", "beinahe", "normalerweise", "zu 99%", "das sollte nie vorkommen"

Wie kann ich den Gefahren aus dem Weg gehen?

Nicht der Kunde muss etwas von Datenbanken und Webapplikation verstehen, sondern *ich* muss in die Welt des Kunden eintauchen.

Ich muss dem Kunden klarmachen, dass die Datenbank "dumm" ist und im Gegensatz zu einem menschlichen Gegenüber die Wünsche nicht von den Augen ablesen *kann* (nicht nur nicht *will*).

Eine Datenbank macht vieles möglich, aber nicht alles. Es ist wichtig, den Kunden rechtzeitig darauf hinzuweisen, welche Anforderung die Datenbank sicher nicht erfüllen können wird.

Wieviel "Technik" darf sein?

10 Gebote zu "Wie vergräme ich den König?"

1. Es gibt nur eine Datenbank und die heißt Oracle
2. Du sollst die Oracle Fachbegriffe nicht verunstalten.
3. Du sollst NOT-NULL-Spalten als solche bezeichnen.
4. Du sollst dem König Constraints an den Kopf werfen.
5. Du sollst "Cascading Delete" erwähnen, damit es den Daten wohlergehe
6. Du sollst von LOV sprechen.
7. Du sollst Trigger und CLOB erwähnen.
8. Du sollst Sessions erklären.
9. Du sollst nach Page und Application Items fragen.
10. Du sollst nicht begreifen deines Königs Wunsch.

Der König will verstehen: Übersetzungsübungen

- 1:n-Beziehung Personen - Orte = ein typischer Fall, wo nur eine Auswahl möglich ist: in einem Ort werden viele Menschen geboren, ein Mensch kann aber nur an einem Ort geboren werden, also nur einen Geburtsort haben
- n:n-Beziehung Personen - Orte: ein Mensch kann im Laufe seines Lebens an verschiedenen Orten wohnen, ein Ort hat immer mehrere Personen als Bürger
- **Url im Browser** = das Feld oben im Firefox / Internet Explorer / Chrome / Safari (je nachdem was im Königshaus benutzt wird), wo die Internet-Adresse steht, die beginnt immer mit http
- **GUI** = Aussehen der Internetseite, = Benutzeroberfläche (für Fortgeschrittene)
- **Drop Down Liste** = Auswahlliste
- **Popop LOV** = eine Auswahlliste öffnet sich in einem neuen kleinen Fenster
- **Report** = Anzeige der Daten

- NOT NULL = Pflichtfeld
- Client-seitig = das geschieht direkt bei der Bedienung Ihres Internet-Programms, nicht in der Datenbank

Der König will mitgestalten: Wie hilft Apex beim "Schauen mit den Händen"?

Begriffe wie Applikation, Seiten, Felder, Auswahllisten werden durch die praktische Erfahrung im wahrsten Sinn des Wortes anschaulicher.

Datumsfelder beispielsweise lassen nur Datum zu, kein "von...bis" oder "um" oder "ca." Das merkt sich der König am besten, wenn er es selbst ausprobieren kann

Feldgrößen können ausprobiert werden. Faustregel: eine durchschnittliche DIN A4_Seite hat 2000-3000 Zeichen.

Die Bedienbarkeit, das "Look and Feel" einer Apex-Applikation kann schon zu Beginn der Applikationsentwicklung vom Kunden getestet und durch entsprechende Rückmeldungen den jeweiligen individuellen Wünschen angepasst werden.

Der König bestimmt die Hilfetexte

Sowohl auf Seitenebene als auch für jedes einzelne Item ist in Apex die Eingabe von Hilfetexten durch den Entwickler vorgesehen. Solide benutzerfreundliche Apex-Applikationen sollten aber aus technischer Hinsicht selbsterklärend und intuitiv sein und daher keine weiteren Hilfetexte erfordern.

Inhaltsbezogene Hilfetexte können hingegen sehr wohl erforderlich sein. Diese Hilfetexte sollten jedoch keinesfalls vom Entwickler erstellt werden. Erstens ist in diesem Fall der Kunde nicht nur König, sondern Kaiser (= Experte), schließlich bezieht sich die Hilfe auf sein Fachgebiet. Zweitens können sich solche Hilfetexte im Lauf der Zeit ändern, erweitert oder auch hinfällig werden. Der Kaiser braucht also die "Hoheit" über seine angebotene Hilfe.

Wie lässt sich dies nun verwirklichen?

- Wir erstellen eine eigene Hilfetabelle HILFETEXTE mit Applikationsnamen, Itemnamen und Hilfetexten:

```
CREATE TABLE helptexts
  (id NUMBER(6,0) NOT NULL,
  title VARCHAR2(200),
  app_id NUMBER,
  app_name VARCHAR2(100),
  page_id VARCHAR2(100),
  item_name VARCHAR2(100),
  text VARCHAR2(4000) ,
  CONSTRAINT hilfetexte_pk
  PRIMARY KEY (id)
  USING INDEX)
```

- Wir befüllen die Tabelle HELPTEXTS mit den in der Applikation vorhandenen Item Names, Item Labels und allenfalls schon vorhandenen Hilfetexten:

```
INSERT INTO helptexts
  (title, app_id, app_name, page_id, item_name, text)
SELECT label, application_id, application_name,
  page_id, item_name, item_help_text
FROM apex_application_page_items
WHERE application_id = my_application_id
```

- ▶ Wir erstellen eine Seite (z.B. P100001) mit einem SQL-Report für einen bestimmten Hilfetext, der sich auf genau eine Seite in einer bestimmten Applikation bezieht:

```
SELECT text
FROM helptexts
WHERE application_id = :P100001_APP_ID
AND item_name IS NULL
AND page_id = :P100001_PAGE_ID
```

- ▶ Wir erstellen eine Seite (z.B. P100002) mit einem SQL-Report für einen bestimmten Hilfetext, der sich auf genau ein Item in einer bestimmten Applikation bezieht:

```
SELECT text
FROM helptexts
WHERE application_id = :P100002_APP_ID
AND item_name = :P100002_ITEM_NAME
```

- ▶ Mit einem entsprechenden Plugin erscheint der Hilfetext wie gewohnt in einem Modal Window:

- Plugin (modal_page.zip) herunterladen:
<http://skillbuilders.com/oracle-apex/Application-Express-APEX-Consulting-Training.cfm?tab=free-plugin-downloads>
- modal_page.zip entpacken: dynamic_action_plugin_com_skillbuilders_modal_page.sql
- In Apex importieren
- Plugin installieren

- ▶ Dynamic Action für Seitenhilfe konfigurieren

Dynamic Action: 1 of 1 Name: F1_GET_PAGE_HELP Set: none

Show All Identification When True Actions False Actions Advanced Condition Authorization Configuration Comments

Identification

Page: 0. P0 Global

* Name: F1_GET_PAGE_HELP

* Sequence: 10

When

* Event: Key Down

* Selection Type: DOM Object

* DOM Object: document

Condition: JavaScript expression

* Value: this.browserEvent.keyCode == 112

Abb. 1: Dynamic Action für Modal Page bei Seitenhilfe erstellen

- 1. True Action:
Execute JavaScript Code
// Standard-Event für F1 in bestimmten Browsern unterdrücken:
this.browserEvent.stopPropagation();
this.browserEvent.preventDefault();
try {
 document.onhelp = function() {return(false);}
 window.onhelp = function() {return(false);}
} catch (e) {}

- 2. True Action:
 Identification:
 Action aus Selectlist Effect auswählen:
 SkillBuilders Modal Page (1.0.0) [Plug-in]
 Settings:
 Dialog Title: Help Text for Page &APP_PAGE_ID.
 Url Location: Statically Defined
 Static URL:
 f?p=&APP_ID.:100001:&APP_SESSION.::::P100001_PAGE_ID:&APP_PAGE_ID.
 Auto-close On Element Selector:
 Dialog Height/Width Mode: Auto

► Dynamic Action für Hilfe zu Item konfigurieren

Dynamic Action: 2 of 2 Name: GET_ITEM_HELP Set: none

Show All Identification When True Actions False Actions Advanced Condition Authorization Configuration Comments

Identification

Page: 0, P0 Global

* Name: GET_ITEM_HELP

* Sequence: 20

When

* Event: Click

* Selection Type: jQuery Selector

* jQuery Selector: [class\$=-w-help]

Condition: - No Condition -

True Actions

The following actions will be fired when the 'When' condition is met, or when it is 'No Condition'.

Edit	Sequence	Action	Fire On Page Load	Selection Type	Affected Elements
	10	SkillBuilders Modal Page (1.0.0) [Plug-in]	No		

1 - 1

Abb. 2: Dynamic Action für Hilfe zu Items mit Modal Page erstellen

- True Action:
 Identification:
 Action aus Select List Effect auswählen:
 SkillBuilders Modal Page (1.0.0) [Plug-in]
 Settings:
 Dialog Title: Hilfe zu Feld (#CURRENT_ITEM# funktioniert nicht)
 Url Location: Attribute of Triggering Element
 Attribute Name: href
 Auto-close On Element Selector:
 Dialog Height/Width Mode: Auto

► Damit der Hilfetext wie gewohnt durch Anklicken des Item Labels aufgerufen werden kann, modifizieren wir die entsprechenden Templates:

```
Template Required with help - Definition: Before Label:  
<label for="#CURRENT_ITEM_NAME#" tabindex="999">  
<a onclick="javascript: return false;" class="required-w-help"  
href="f?p=&APP_ID.:100002:&APP_SESSION.::::  
P100002_ITEM_NAME:#CURRENT_ITEM_NAME#" tabindex="999" >  

```

Template *Optional with help* - Definition: analoge Vorgehensweise

► Nun können wir den Report für den Hilfetext im Modal Window wie gewohnt darstellen. Den vom Kunden gewählten Titel können wir als Überschrift verwenden.

Erledigt!

Kontaktadresse:

Dr. Ingrid Hayek
daten & worte
Höhenstraße 110
A-6020 Innsbruck

Telefon: +43 (0) 650-2923290
E-Mail work@ingridhayek.at
Internet: ingridhayek.at