

**MEHR ALS
HEADER UND HANDLER...**

**SYSTEM
DESIGN
TOOLKIT**

**...VOM USE CASE
ZUM CODE**

Copyright © 2006 Stefan Sachs

UML ist ein eingetragenes Warenzeichen von Object Management Group, Inc. KORG ist ein eingetragenes Warenzeichen von Korg Inc. Visual Age ist ein eingetragenes Warenzeichen von IBM Corp. PARTS ist ein eingetragenes Warenzeichen von Digitalk Inc. Bean Builder ist ein eingetragenes Warenzeichen von Sun Microsystems, Inc. Alle übrigen Produkte und Markennamen, die in dieser Broschüre genannt werden, sind eingetragene Warenzeichen der entsprechenden Rechtsinhaber.

Die Abbildung eines KORG Chromatic Tuner CA-20 erfolgt mit freundlicher Genehmigung von Korg Inc. Es sei an diese Stelle ausdrücklich darauf hingewiesen, daß das hier vorgestellte Entwurfswerkzeug nicht bei der Entwicklung dieses Geräts verwendet wurde.

Von der technischen Lösung zum Produkt

Software ist ein entscheidender Faktor für den Erfolg eines Produkts geworden. Wo noch vor wenigen Jahren eingestellt und geschaltet wurde, wird inzwischen über Touchscreens, Scrollräder oder Wippen aus Menus ausgewählt. Wenn die Bedienungsanleitung fehlt, dann sind viele Geräte nutzlos; ältere Personen sind hilflos.

Ihr Produkt hat Funktionen, die der Kunde nutzen will. Für den Verkaufserfolg ist es entscheidend, daß eine gut gestaltete Bedienoberfläche dem Kunden den Zugang zu den Möglichkeiten Ihres Produkts öffnet.

Wenn dieses Potential erschlossen wird, dann entsteht daraus ein echter Wettbewerbsvorteil. Produkte wie iPod zeigen, daß die Bedienoberfläche eine entscheidende Rolle für den Markterfolg spielt.

Die Bedienoberfläche ist entscheidend für den Markterfolg

Wie entstehen bessere Bedienoberflächen?

Das Gestalten der Bedienung eines Geräts ist keine Tätigkeit, die ‚so nebenbei‘ erledigt wird.

Das Äußere des Geräts muß seiner Verwendung angepasst sein; Bedienelemente müssen da sein, wo die Hand des Benutzers sie erwartet; Anzeigen dort, wo sein Blick zuerst hinfällt.

Bedienabläufe sollen keine Überraschungen bergen.

Umfang und Gestaltung der Abläufe sollen den Benutzer nicht überfordern, aber auch das volle Potential der Möglichkeiten des Geräts erschließen.

Die besten Bedienoberflächen entstehen, wenn die Entwicklung im Dialog mit dem Anwender erfolgt. Ein Werkzeug, das Ihre Ideen sichtbar und begreifbar macht, kann hier den entscheidenden Beitrag zum Erfolg liefern.

Die Gestaltung der Bedienung ist eine anspruchsvolle Aufgabe, das richtige Werkzeug kann wesentlich zum Erfolg beitragen

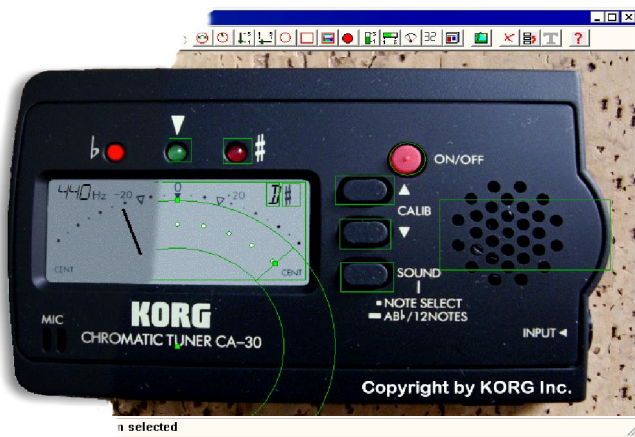
Design

Mit dem GUI Simulator erstellen Sie in kurzer Zeit realistische Simulationen

Der erste Eindruck von Ihrem Produkt wird über die Gestaltung der Bedienelemente vermittelt. Es ist deshalb wichtig, daß eine 'begreifbare' Anordnung der Bedienelemente in einer realistischen Darstellung von Anfang an in die Entwicklungsarbeit einbezogen wird.

Während bei Anwendungsprogrammen eine einheitliche Bildschirmausgabe über Maus und Tastatur angesprochen wird, sind in Geräten häufig heterogene Anzeigeelemente (LEDs, Siebensegmentanzeigen, gelegentlich sogar Zeigerinstrumente) vorhanden, die Eingabe erfolgt über Tasten, Drehknöpfe, gelegentlich auch berührungsempfindliche Anzeigen.

Ein GUI Builder, wie er zu vielen Entwicklungsumgebungen mitgeliefert wird, ist deshalb nicht geeignet, die Bedienoberfläche nachzubilden. Unser Werkzeug geht einen anderen Weg um realistische Simulationen schnell zu erstellen: Eine eingescannte Zeich-



nung, eine Photographie oder eine aus dem CAD System exportierte Bitmap läßt sich mit einem Baukasten aus maussensitiven Flächen, Bitmap- und Vektorgraphiken in eine bedienbare Simulation verwandeln. Displayinhalte können ebenso dargestellt werden, wie reine Hardware-Anzeigeelemente (LEDs, Sieben Segment Anzeigen).

Handlungsabläufe

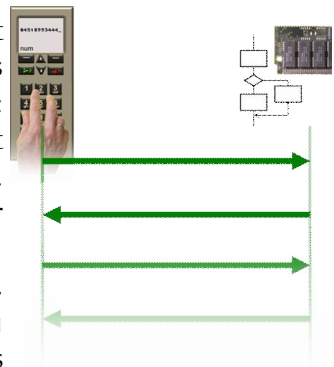
Handlungsabläufe organisieren die Interaktion zwischen Benutzer und Gerätefunktion.

Auf der einen Seite ist der Kunde, der etwas von Ihrem Produkt will - auf der anderen Seite ist das Produkt, das auf eine Anforderung wartet, um dann etwas Nützliches für den Kunden zu tun. Sie kennen genau die Sprache, die Ihr Produkt versteht. Je anspruchsvoller die Funktion Ihres Produkts ist, desto weniger wahrscheinlich ist es, daß auch der Kunde die Sprache Ihres Produkts versteht. Die Bedienoberfläche ist der Dolmetscher zwischen Ihrem Kunden und Ihrem Produkt. Damit das funktioniert, muß die Bedienoberfläche die Sprache des Kunden sprechen und verstehen.

Das Vokabular, das wir im Umgang mit Geräten verwenden ist, natürlich nicht das Vokabular, das wir im Alltag verwenden; dennoch haben sich auch im Umgang mit Gegenständen ‚Sprachen‘ herausgebildet. Sprechen dieser Sprachen heißt, Tasten oder Einsteller des Geräts in der richtigen Reihenfolge zu betätigen; Zuhören heißt, Anzeigen abzulesen, auf Kontrolleuchten zu achten, auf Warntöne zu hören. So wie es uns leichter fällt, Sprachen zu benutzen, die wenige Vokabeln haben und deren Satzbau einfach und regelmäßig ist, genauso wird uns der Umgang mit Geräten leichter fallen, wenn die Befehlsfolgen knapp und eingängig sind und die Reaktionen des Geräts erwartet und vorhersehbar sind.

Unser Produkt hilft Ihnen dabei, eine Bedienoberfläche zu erstellen, die die Funktionen Ihres Produkts perfekt in die Sprache Ihrer Kunden übersetzt.

**Die
Arbeitsabläufe
Ihrer Kunden
zu verstehen ist
der Schlüssel zu
einer flüssigen
Bedienung**



Use Cases, Dialoge zwischen Benutzer und System

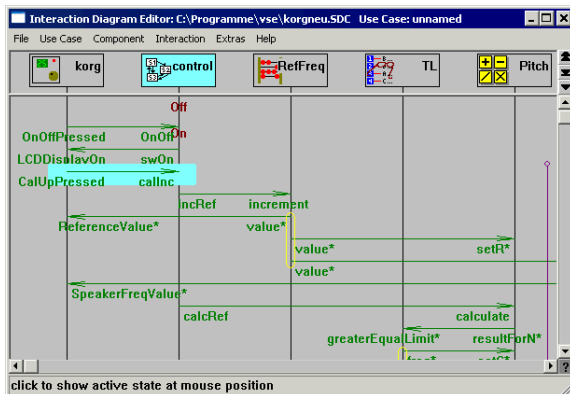
Beschreibungen von Arbeitsabläufen sind die Entwurfsgrundlage

Ein Use Case beschreibt einen vollständigen Ablauf einer Bedienung.

Grundlage aller weiteren Schritte ist eine Sammlung von solchen Use Cases, die alle Aspekte der Benutzung Ihres Produkts umfaßt.

Use Cases können in vielen Formen erfaßt werden. Die am meisten verbreitete Methode, frei als Text formulierte Beschreibungen, ist zugleich die am wenigsten nützliche. Der fehlende Zwang zur Exaktheit, der natürlicher Sprache zu eigen ist, führt dazu, daß die Formalisierung an Mehrdeutigkeiten und inkonsistenten Abstraktionsebenen scheitert.

Es ist also sinnvoll, die Ausdrucksmittel für die Beschreibung



von Use Cases zu beschränken. Bei der Bedienoberfläche bietet sich hier an, nur Aktionen zuzulassen, die von den Bedien- und Anzeigeelementen unterstützt werden; d.h. eine Taste kann gedrückt oder losgelassen werden, eine LED kann an oder aus sein oder sie kann blinken.

Bei dem zu bedienenden System muß ebenfalls eine eindeutige Schnittstellendefinition vorhanden sein; Ein- und Ausgaben müssen in einem Protokoll definiert sein.

Mit diesen Restriktionen wird eine einheitliche Abstraktionsebene erzwungen; Bedienabläufe werden stets bis auf elementare Aktionen aufgelöst. Dies mag im ersten Schritt mehr Arbeit bedeuten, die eindeutige Darstellung zahlt sich jedoch im weiteren Verlauf vielfach aus.

Das Werkzeug unterstützt Sie dabei, exakt zu formulieren

Unser Werkzeug bietet drei Verfahren, die Eingabe von Use Cases zu unterstützen:

- visuelle Eingabe in Interaktionsdiagrammen
- Erfassung der Benutzereingaben an der Simulation und Auswahl der Systemreaktionen im Dialog
- verbale Beschreibung durch einen syntaxgesteuerten Editor

Alle drei Verfahren erzeugen formal korrekte Eingaben, die in den weiteren Schritten genutzt werden.








Freie Auswahl der Art der Erfassung

Komponenten

In jedem Systementwurf gibt es typische Aufgaben, für die bewährte Lösungen existieren. Es ist Zeitverschwendung, diese Lösungen jedes Mal neu zu erstellen. Für solche Auf-

Fertige, wiederverwendbare Komponenten verkürzen die Entwicklungszeit

Standardkomponenten

| | | |
|---|--------------------|--|
|  | Counter | Einfacher Zähler |
|  | Calculation | Numerische Berechnungen |
|  | Comparison | Vergleiche zwischen numerischen Werten |
|  | Timer | Einfacher Timer |
|  | Table Lookup | In Tabelle suchen oder Cursor bewegen |
|  | Monitored Value | Zahlenwerte gegen Grenzen überwachen |
|  | Alphanumeric Entry | Texteingabe mit Cursor |

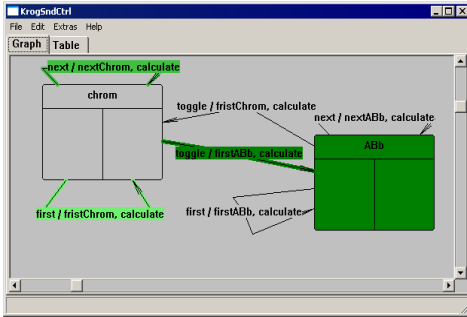
gaben stehen in dem Werkzeug fertige Komponenten zur Verfügung. Die Anpassung an die aktuellen Anforderungen erfolgt über einen Konfigurationsdialog.

Zustandsautomaten

Viele Bedienabläufe erfolgen in mehr als einem Schritt, d.h. das System kennt verschiedene Zustände. Aktionen des Benutzers haben, abhängig vom Systemzustand eine andere Bedeutung. So kann ein Drehknopf zur Auswahl eines Stellwerts dienen, nach der Auswahl verändert er den gewählten Wert.

Komplexe Abläufe einfach gestalten

Die Gestaltung solcher mehrstufiger Abläufe ist entscheidend für die Benutzerfreundlichkeit des Systems: Klare Strukturen, verständliche, eindeutige Rückmeldungen und einheitliche Sequenzen erleichtern es, auch komplexe Aufgaben ohne Handbuch zu bewältigen.



Zustandsautomaten sind ein seit langem bewährtes Verfahren zur Beschreibung solcher Abläufe. Einfach und robust haben sie sich in unzähligen Implementierungen mit höchsten Sicherheitsanforderungen bewährt.

Ein Editor für Zustandsautomaten gehört zu diesem Werkzeug.

Sowohl die graphische Darstellung als auch die Tabellenform werden unterstützt. Wenn die gesammelten Use Cases die möglichen Abläufe hinreichend abdecken, dann können Zustandsautomaten vollständig generiert werden. Ebenso ist es möglich, Use Cases anhand der Zustandsautomaten durch die Vorgabe von Ereignissen zu erzeugen.

Automatische Erzeugung von Zustandsautomaten aus den gesammelten Use cases

Simulation und Visualisierung

Ausführbare Modelle

Das auf diese Weise erzeugte Modell stellt eine funktionsfähige Simulation dar. Auf dem Bildschirm lässt sich das System bedienen, alle Abläufe können in den Visualisierungen verfolgt werden. Selbst schnell ablaufende Vorgänge bleiben dank der Time Fader Funktion (die letzten Abläufe werden mit verblassenden Farbmarkierungen hinterlegt) nachvollziehbar. Durch die konsequente Integration der verschiedenen Diagramme, die umfangreichen Konsistenzprüfungen und die intuitiven Editiermöglichkeiten lassen sich mit dem Modell Varianten mit geringem Zeitaufwand erproben.

Schnelle Änderungen, viel Spielraum für Erprobung

Die Diagramme orientieren sich an den Konventionen der UML, sind anschaulich und liefern eine umfassende Systemdokumentation.

Stets aktuelle Dokumentation

Für Kundenbefragungen können unabhängig vom Werkzeug lauffähige Simulationen erstellt werden; Eingaben der Benutzer können aufgezeichnet und später ausgewertet werden.

Simulationen unterstützen den Kundendialog

Warum Codegenerierung so wichtig ist

Ein Entwurfswerkzeug ist immer hilfreich, um Systeme zu gestalten. Dem steht aber stets ein nicht unbeträchtlicher Lernaufwand für das Werkzeug und die von dem Werkzeug unterstützten Verfahren gegenüber. Wie groß der Nutzen insgesamt ist, das hängt davon ab, wie effizient und nachhaltig das Wissen, das in dem Entwurf gesammelt wurde, genutzt wird.

Dieses Werkzeug erzeugt aus dem erstellten Modell vollständigen, lesbaren und effizienten Code für das Zielsystem. Das ermöglicht Ihnen, Ihr Produkt auf Modellebene zu optimieren, weiterzuentwickeln und zu pflegen. Die Turnaround Zeiten für eine neue Version werden auf Minuten reduziert, die Codegenerierung erfolgt in Sekunden.

Das Modell wird komplett in Code für das Zielsystem umgesetzt

Mit Rapid Prototyping zu besseren Produkten

So wird es möglich, sobald die Zielhardware zur Verfügung steht, Änderungen unmittelbar nach der Definition im realen Gerät zu erproben. Das eröffnet ganz neue Möglichkeiten, die Ergonomie eines Produkts zu optimieren, es entsteht ein ganzheitlicher Ansatz, der Haptik und Anmutung in den Entwicklungsprozess einbezieht.

Kurze Entwicklungszyklen, mehr Projektsicherheit

Zur Technik

Das Gesamtsystem besteht aus mehreren Werkzeugen, die eng miteinander verknüpft sind:

- Der **Editor für Interaktionsdiagramme** ist zugleich das zentrale Fenster. Hier werden die Komponenten des Systems verwaltet und die Use Cases bearbeitet. Die Darstellung der Interaktionsdiagramme orientiert sich an der UML, wurde aber um ein Event-Modell erweitert.
- Der **Simulator für Benutzerschnittstellen** erzeugt realistisch animierte, bedienbare Modelle.
- Der **Editor für Zustandsautomaten** unterstützt sowohl Graphen als auch Tabellen. Simulationsabläufe werden über mehrere Schritte sichtbar gemacht. Die Darstellung entspricht UML 1.5, State Charts werden bewusst nicht unterstützt; mehrfache Zustände werden stattdessen durch verknüpfte Zustandsautomaten dargestellt.
- Der **Editor für Kollaborationsdiagramme** kennt zwei Modi: Ein einzelner Use Case kann als UML Kollaborationsdiagramm dargestellt werden oder es können alle Verbindungen zwischen Komponenten des Systems bearbeitet werden. In diesem Modus entspricht der Editor einer visuellen Workbench, wie man sie von Visual Age, Parts oder Sun's BeanBuilder kennt.
- Der **Editor für Verbindungen** dient ebenfalls dazu, die Verbindungen zwischen Komponenten zu bearbeiten, allerdings nutzt er Auswahllisten was bei großen Modellen übersichtlicher sein kann als die graphische Darstellung.
- Der **Code Generator** dient dazu, die Erzeugung von Code für das Zielsystem zu steuern. Er vereinigt Werkzeuge für die Anpassung an die Hard- und Softwarearchitektur des Zielsystems mit einer einfachen Änderungskontrolle.
- Das **Laufzeitsystem** ermöglicht es, Simulationen ohne die Entwicklungswerkzeuge ablaufen zu lassen. Es können selbstablaufende Simulationen erstellt werden, die Benutzereingaben im Hintergrund aufzeichnen

Systemanforderungen:

Windows 9X, NT oder XP

Pentium III, 1 GHz,

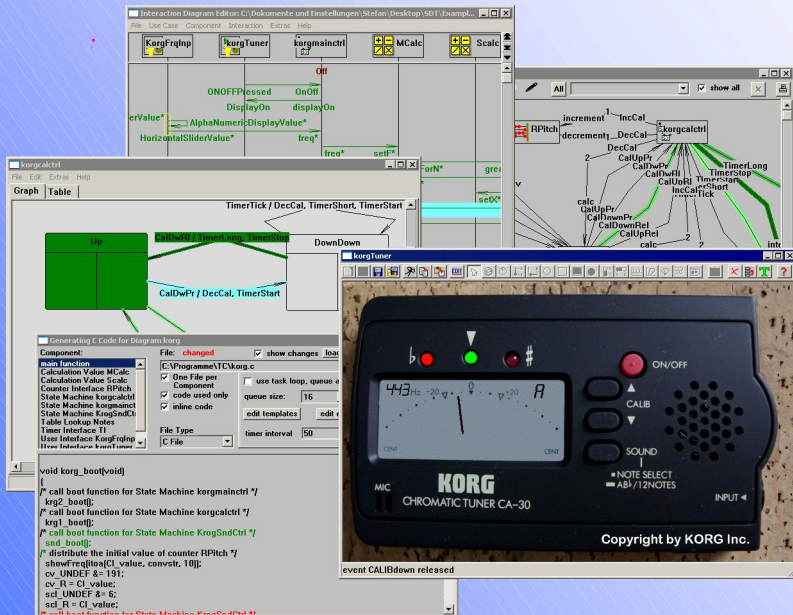
mindestens 256 MB RAM,

mindestens 100 MB freie Plattenkapazität,

Maus mit Scrollrad,

24 bit Grafik, Auflösung min. 1024 x 768

Mehr Bildschirme und bessere Auflösung erleichtern die Benutzung des Systems



Stefan Sachs
Dr. Ing.
Beratender Ingenieur

Ringreiterweg 20
23558 Lübeck

Tel. +49 (451) 8993444
Fax +49 (451) 8993445
E-Mail: ssachs@acm.org
web: www.ssachs.de