

Ausarbeitung zum Thema Softwareergonomie

Inhaltsverzeichnis

1	Was ist Softwareergonomie ?	2
2	Definition von Gebrauchstauglichkeit (aus der ISO 9241)	2
2.1	Effektivität	2
2.2	Effizienz	3
2.3	Benutzerzufriedenheit	3
3	Wann ist eine Software ergonomisch ?	4
4	Grundsätze ergonomischer Dialoggestaltung	4
4.1	Was ist ein Dialog?	4
4.2	Aufgabenangemessenheit	5
4.3	Selbstbeschreibungsfähigkeit	5
4.4	Steuerbarkeit	6
4.5	Erwartungskonformität	7
4.6	Fehlerrobustheit	7
5	Softwareergonomie in der Praxis	8
5.1	Vorstellung des Projektes	8
5.2	Schematischer Aufbau	9
5.3	Das Userinteface	10
5.4	Schlussbemerkung	13
6	Quellen und Literaturhinweise	13

1 Was ist Softwareergonomie ?

Die Softwareergonomie ist ein interdisziplinäres Forschungsgebiet mit einem sehr praxisbezogenen Anwendungsbereich. Untersucht werden sämtliche Fragestellungen im Bereich der Mensch-Maschine-Interaktion.

Das Ziel ist es, Computersysteme zu entwickeln, die eine problemlos und intuitiv anwendbare Funktionalität besitzen, die einfach zu erlernen und zu bedienen sind. Anwendungen sollten so gestaltet sein, dass der Benutzer intuitiv versteht, wie er mit dem Programm arbeiten kann.

„*Usability*“ steht für die Brauchbarkeit und die Bedienungsfreundlichkeit von technischen Geräten, welche von Menschen bedient werden. Solche Geräte bezeichnet man als interaktive Systeme, weil sie eine Benutzerschnittstelle aufweisen. Typische Benutzerschnittstellen für einen Computer sind die Tastatur, die Maus, der Bildschirm, das installierte Betriebssystem und die verwendete Software.

Der Begriff „*Usability*“ bedeutet, dass ein System leicht erlernbar und effizient benutzbar ist, eine hohe Fehlertoleranz aufweist und Benutzerzufriedenheit bewirkt.

Computersysteme welche den oben genannten Anforderungen genügen, ermöglichen es den Anwendern, ihre Aufgaben produktiv und problemlos zu erledigen

- **sie sind ergonomisch.**

2 Definition von Gebrauchstauglichkeit (aus der ISO 9241)

Allgemein kann *ergonomisch* mit *menschengerecht* übersetzen werden. Bei dem Werkzeug Software nimmt man sinnvoller Weise auch die Aufgabe hinzu, die mit dem Werkzeug bearbeitet werden soll und erhält so den schon unter Punkt 1 genannten Begriff *Gebrauchstauglichkeit (=Usability)*.

Diese ist in einer internationalen Norm – der ISO 9241 - als das Produkt aus Effektivität, Effizienz und Zufriedenstellung definiert:

2.1 Effektivität

Es muss sichergestellt sein, dass der Benutzer die vorgesehenen Aufgaben mit der Software erledigen kann und die benötigten Ergebnisse korrekt erreicht werden.

Beispielsweise muss eine Übersetzungssoftware am Ende eine *sachlich richtige* Übersetzung des gewünschten Begriffes geben.

In der Regel wird dieser Punkt bereits durch die Tests während der Herstellung der Software abgedeckt. In der ISO 12119 sind diese „klassischen“ Softwarequalitätseigenschaften rund um die Korrektheit abgedeckt.

2.2 Effizienz

Ein Werkzeug macht nur dann Sinn, wenn es die Arbeit erleichtert. Software muss also nicht nur *benutzbar* sondern auch *nützlich* sein.

Die Arbeitserleichterung hängt wesentlich von der Senkung des notwendigen Bedienaufwandes insbesondere der aufzuwendenden Denkarbeit („Wie ging das noch?“, „Wo war noch die Funktion XY?“) ab. So kann aufgrund von Bedienbarkeitsmängeln ein beträchtlicher Teil der Arbeitszeit „verloren“ gehen. An dieser Stelle bemerkt man den wirtschaftlichen Effekt einer menschen- und aufgabengerechten Gestaltung von Software.

Nach einer Studie der englischen Beratungsfirma „System Concepts Ltd.“ gehen 60% aller Probleme auf eine nicht aufgabengerechte Gestaltung der Software zurück.

2.3 Benutzerzufriedenheit

Die Zufriedenheit der Benutzer bei der Benutzung der Software war wohl ausschlaggebend für die Bezeichnung „Benutzerfreundlichkeit von Software“. Dabei entsteht irrtümlich der Eindruck, dass es sich um eine „nice-to-have“ Eigenschaft handelt, die man gerne für sein Produkt beansprucht, aber die nicht von wirklicher wirtschaftlicher Bedeutung ist.

Abgesehen davon, dass es eine arbeitspsychologische Binsenweisheit ist, dass unzufriedene Menschen auch weniger produktiv sind, so ist die Benutzerunzufriedenheit auch ein wesentlicher Indikator für eine nicht aufgabengerechte Gestaltung des Werkzeuges Software.

Eine mangelnde Zufriedenheit bei den Benutzern ist häufig (aber nicht immer) ein Indiz für Gebrauchstauglichkeitsprobleme der Software. Mögliche andere Auslöser sind Probleme bei der Aufgabengestaltung (Unterforderung oder Überforderung) und der Umgebungsbedingungen (Raumgestaltung, Beleuchtung, Möbel aber auch Schulung, organisatorische Mängel).

Umgekehrt gibt die subjektive Zufriedenheit der Benutzer keine Gewähr für ein ergonomisch gutes Produkt. Viele Mängel (zum Beispiel wiederholte Abstürze, fehlende Funktionen oder komplizierte Bedienung) werden aus organisatorischen

Gründen (vielleicht aus Angst um den Arbeitsplatz) nicht bemängelt oder mangels Vergleich gar nicht bewusst wahrgenommen. Die meisten Benutzer haben sich zum Beispiel scheinbar daran gewöhnt, ihren Rechner von Zeit zu Zeit neu starten zu müssen. Viele Probleme werden auch aufgrund schlechter Analysemethoden (z.B. ungeschickte Befragungen) nicht entdeckt.

3 Wann ist eine Software ergonomisch ?

Kurz gesagt ist eine Software dann gebrauchstauglich wenn sie an den Benutzer und dessen Aufgaben angepasst ist und für ihn effektiv, effizient und ohne Unzufriedenheit zu benutzen ist.

Prinzipiell sind dabei die Hauptaufgaben und die wesentlichsten Benutzereigenschaften zu betrachten. Wenn der Benutzer durch das Werkzeug wesentlich beeinträchtigt wird (zum Beispiel durch fehlende Unterstützung, Mehraufwand oder fehlerhafte Ergebnisse), dann ist die Software nicht mehr ergonomisch im Sinne der Bildschirmarbeitsverordnung.

4 Grundsätze ergonomischer Dialoggestaltung

4.1 Was ist ein Dialog?

Wenn man von einem Dialog im Alltag spricht, so ist dort in erster Linie der Austausch von Informationen im Sinne einer Konversation von Mensch zu Mensch gemeint. Der Informationsaustausch findet vorrangig über das Medium Sprache und Gestik statt. Stichwort Kommunikation.

Um erfolgreich mit einer Maschine kommunizieren zu können, bedarf es ebenfalls eines Dialogs – dem Mensch-Maschine-Dialog. Dieser kann sehr unterschiedlich gestaltet sein, aber immer mit dem selben Ziel: dem Informationsaustausch. So benötigt der Mensch von der Maschine (z.B. dem Computer) ein Feedback über das was er gerade macht. Die Maschine benötigt ihrerseits Informationen vom Anwender, um ihre Arbeit aufnehmen zu können und so in der Lage ist, für den Anwender brauchbare Ergebnisse zu erzielen.

Bei Software findet der Informationsaustausch überwiegend manuell und visuell statt. Der Anwender übergibt seine Informationen mittels Eingabegeräte wie Tastatur, Maus,

Lightpen, Barcodescanner seltener auch Mikrophon an die Software. Die Software bereitet die oft sehr komplexen Daten zu „überschaubaren“ Informationen für den Anwender auf. Je nach Zweck gibt sie diese in Form von Zahlen, Texten, grafischen Darstellungen, akustischen Signalen oder gar per Sprache aus.

In den folgenden Abschnitten soll nun näher auf die Anforderungen der Mensch-Maschine Dialoggestaltung eingegangen werden.

4.2 Aufgabenangemessenheit

Wann ist ein Dialog aufgabenangemessen?

Ein Dialog ist aufgabenangemessen, wenn er die Erledigung der Arbeitsaufgabe des Benutzers unterstützt, ohne ihn durch Eigenschaften des Dialogsystems unnötig zu belasten.

Tätigkeiten, die sich aus der technischen Eigenart des Dialogsystems ergeben, durch das System selbst ausgeführt werden.

Der Dialog soll den Arbeitsaufgaben angepasst sein. Bei der Dialoggestaltung sind insbesondere deren Komplexität sowie Art und Umfang der Information, die der Benutzer zu verarbeiten hat zu berücksichtigen. Art und Form der Eingabe sollen der Arbeitsaufgabe angepasst sein.

Die im Dialog verfügbaren Arbeitsmittel sollten vom Benutzer an regelmäßig wiederkehrende Arbeitsaufgaben angepasst werden können.

Existieren für eine Arbeitsaufgabe Eingabevorüberlegungen (z.B. Werte, die standardmäßig vorgegeben sind), sollen dem Benutzer die Eingabe solcher Werte erspart werden. Vorgabewerte sollen durch andere Werte oder durch andere Vorgabewerte geändert werden können.

Wenn es für Vergleichszwecke erforderlich ist, sollen während der Erledigung einer Arbeitsaufgabe, bei der Daten geändert werden, die ursprünglichen Daten erhalten bleiben und wieder abrufbar sein. Dies kann auch für fehlerhafte und unvollständige Daten gelten.

4.3 Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn dem Benutzer auf Verlangen Einsatzzweck sowie Leistungsumfang des Dialogsystems erläutert werden können und wenn jeder einzelne Dialogschritt verständlich ist oder der Benutzer auf Verlangen dem jeweiligen Dialogschritt entsprechende Erläuterungen erhalten kann.

In Ergänzung zur Benutzerschulung sollen diese Erläuterungen dazu beitragen, dass sich der Benutzer für das Verständnis und für die Erledigung der Arbeitsaufgabe

zweckmäßige Vorstellungen von dem Systemzusammenhang machen kann; z.B. über Umfang, Aufgaben, Aufbau und Steuerbarkeit des Dialogsystems, über Benutzung dieser Erläuterungen, über Umgang mit Fehlermeldungen.

Erläuterungen sollen an die allgemeinüblichen Kenntnisse der zu erwartenden Benutzer angepasst sein (z.B. in deutscher Sprache, Verwendung beruflicher Fachausdrücke). Dem Benutzer sollen nach Art und Umfang verschiedene Erläuterungen verfügbar sein. Beschreibungen sollen situationsabhängig gegeben werden, um ihren Wert für den Benutzer zu erhöhen.

4.4 Steuerbarkeit

Ein Dialog ist steuerbar, wenn der Benutzer die Geschwindigkeit des Ablaufs sowie die Auswahl und Reihenfolge von Arbeitsmitteln oder Art und Umfang von Ein- und Ausgaben beeinflussen kann. Der Benutzer soll die Geschwindigkeit des Dialogs an seine individuelle Arbeitsgeschwindigkeit anpassen können; z.B. keinen Arbeitstakt, kein Treiben des Benutzers durch Erlöschen von Bildschirmanzeigen. Die Eingabetätigkeit des Benutzers soll nicht durch unnötiges Warten auf die Ausgabe von Daten vorangegangener Dialogschritte aufgehalten werden.

Soweit es die Arbeitsaufgabe erlaubt, soll der Benutzer während des Dialogs die Arbeitsmittel frei wählen und den Arbeitsweg frei bestimmen können. Der Benutzer soll in für ihn leicht überschaubaren Dialogschritten vorgehen können; eine Zusammenfassung von Dialogschritten sollte dem Benutzer ermöglicht werden, wenn es von der Arbeitsaufgabe her sinnvoll ist.

Der Benutzer soll den Dialog jederzeit unterbrechen können, soweit es die Arbeitsaufgabe zulässt. Nach einer Unterbrechung soll er (z.B. aufgrund von Zwischenergebnissen) entscheiden können, ob der Dialog an der Unterbrechungsstelle fortgeführt werden soll.

Ist der Dialog wegen eines Systemausfalls unterbrochen worden, so ist ein Wiederaufnahmepunkt definiert durch den letzten Dialogschritt vor dem Systemausfall; es kann zweckmäßig sein, eine Wiederaufnahme zu einem weiter zurückliegenden Dialogschritt festzulegen. In jedem Fall ist der Wiederaufnahmepunkt bei Fortsetzung des Dialogs mitzuteilen.

Der letzte Dialogschritt soll, soweit seine Folgen reversibel sind und die Arbeitsaufgabe es zulässt, zurückgenommen werden können. Die „Rücknehmbarkeit“ mehrerer zusammenhängender Dialogschritte ist wünschenswert. Die Rücknahme muss in einen Zustand zurückführen, der dem vor dieser Folge entspricht.

Der Benutzer sollte die Menge der darstellbaren Informationen steuern können, soweit es von der Arbeitsaufgabe her sinnvoll ist.

4.5 Erwartungskonformität

Ein Dialog ist erwartungskonform, wenn er den Erwartungen der Benutzer entspricht, die sie aus Erfahrungen mit bisherigen Arbeitsabläufen oder aus einer Benutzerschulung mitbringen sowie den Erfahrungen, die sie sich während der Benutzung des Dialogsystems und im Umgang mit dem Benutzerhandbuch bilden.

Das Dialogverhalten innerhalb eines Dialogsystems soll einheitlich sein. Uneinheitliches Dialogverhalten würde den Benutzer zu starker Anpassung an wechselhafte Durchführungsbedingungen seiner Arbeit zwingen, das Lernen erschweren und unnötige Belastungen mit sich bringen.

Der Benutzer soll Erwartungen hinsichtlich seines Arbeitsablaufs aufgrund der Rückmeldungen des Dialogsystems bilden können. Die für die Führung des Dialogs relevanten Zustandsänderungen des Systems sind dem Benutzer mitzuteilen. Für Eingabedaten in Kurzform oder in verschlüsselter Form soll die Eingabe in Klartext bestätigt werden, wenn die Gewissheit des Benutzers über die Richtigkeit seiner Eingabe verstärkt werden soll. Für besonders schutzbedürftige Eingabedaten kann die Rückmeldung in Klartext entfallen.

Eingegebene Zeichen sollen im allgemeinen unmittelbar auf dem Bildschirm angezeigt werden, um die für die Eingabe erforderliche Aufmerksamkeit nicht durch verzögerte Anzeige zu belasten. Ebenso sollen Positionierungen (z.B. Bewegungen von Positionsmarken) unmittelbar erfolgen.

Der Benutzer macht im Laufe der Benutzung Erfahrungen mit der Antwortzeit eines Dialogsystems und bildet sich daraus Erwartungen. Bei ähnlichen Vorgängen, insbesondere solchen mit kurzer Verarbeitungsdauer, sollen die Antwortzeiten ähnlich sein; bei kalkulierbaren, deutlichen Abweichungen von der üblichen Antwortzeit (z.B. aufgrund einer verlängerten Verarbeitungszeit) soll der Benutzer hiervon unterrichtet werden.

Der Benutzer sollte über den Stand der Bearbeitung informiert werden, damit seine Aufmerksamkeit nicht unnötig gebunden wird. Wird der Dialog aus technischen Gründen unterbrochen, sollen dem Benutzer soweit wie möglich Art, Umfang und Dauer des Ausfalls mitgeteilt werden.

4.6 Fehlerrobustheit

Ein Dialog ist fehlerrobust, wenn trotz erkennbar fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis mit minimalem oder ohne Korrekturaufwand erreicht wird. Dazu müssen dem Benutzer die Fehler zum Zwecke der Behebung verständlich gemacht werden.

Eingaben des Benutzers dürfen nicht zu undefinierten Systemzuständen oder Systemzusammenbrüchen führen können. Es kann zweckmäßig sein, eindeutig korrigierbare Fehler automatisch zu korrigieren und den Verarbeitungsprozess

fortzusetzen. Hierbei ist es im allgemeinen sinnvoll, dem Benutzer die Durchführung der Korrektur in einer für ihn verständlichen Form mitzuteilen. Das automatische Korrigieren muss abschaltbar sein.

Wenn ein Fehler auf verschiedene Weise vom System behoben werden kann, sollten dem Benutzer Korrekturalternativen zur Auswahl angeboten werden, ohne die Möglichkeit zur Neueingabe auszuschließen.

Bei allen Fehlermeldungen soll auch auf den Ort des Fehlers hingewiesen werden (z.B. durch Markierung der Fehlerstelle). Die zur Aufmerksamkeitsregung erforderliche Codierungsform sollte der Anwendung angemessen sein (siehe DIN 66 234 Teil 5). Es kann zweckmäßig sein, Fehlermeldungen unterdrückbar zu machen. Es ist zweckmäßig, zusätzliche Erläuterungen zur Fehlerbehebung vorzusehen.

In der Regel sollen Fehlermeldungen sofort ausgegeben werden. Wenn es der Arbeitsablauf erlaubt, dürfen auf Anforderung des Benutzers Fehlermeldungen später ausgegeben werden, um seinen Denkprozess nicht unnötig zu stören.

Fehlermeldungen sind verständlich, sachlich und konstruktiv zu formulieren. Sie sollten einheitlich strukturiert werden (z.B. nach Fehlerart, möglichen Fehlerursachen und Möglichkeiten der Fehlerberichtigung). Fehlermeldungen dürfen keine Werturteile enthalten, wie z.B. "Unsinnige Eingabe".

5 Softwareergonomie in der Praxis

5.1 Vorstellung des Projektes

Die im folgenden beschriebene Software wurde von mir im Rahmen des Praxissemesters entwickelt, das ich im Bionegineeringlab der University of Utah absolviert habe.

Herr Prof. PhD Dwayne Westenskow, Direktor des Departments of Anesthesiology, gab mir die Möglichkeit, in seinem lab an einem sehr interessanten Projekt zu arbeiten – der Entwicklung einer Software zur Steuerung eines Lungensimulators. Der Lungensimulator stellt das elektrisch/mechanische Äquivalent zur menschlichen Lunge dar. Er wird für die Entwicklung neuer Anästhesiegeräte, Verifizierung von Beatmungsgeräten und Trainingszwecke benötigt.

Die Aufgabe: Das Userinterface „LungSim“ ist in objektorientierter Programmiersprache C++ zu schreiben. Als Entwicklungsumgebung dient hierzu C++ Builder 3. Die Kommunikation mit dem Board (PIC 16C74; D/A- & A/D Wandlern) des Lungensimulators und den anderen Peripheriegeräten soll über RS232 – Schnittstellen realisiert werden.

Zur Visualisierung der Simulationsdaten soll LungSim über eine grafische Oberfläche verfügen, die es dem Benutzer auch erlaubt während einer Simulation die Parameter zu ändern.

Für spätere Auswertungen, Dokumentationen, Weiterverarbeitung und Archivierung der Simulationsdaten ist ein Datenrekorder in LungSim zu implementieren.

Soweit die Aufgabenstellung. Hier soll nun nicht auf die Lösung der technischen Anforderungen, sondern dem Thema entsprechend auf die Gestaltung der Benutzeroberfläche eingegangen werden.

5.2 Schematischer Aufbau

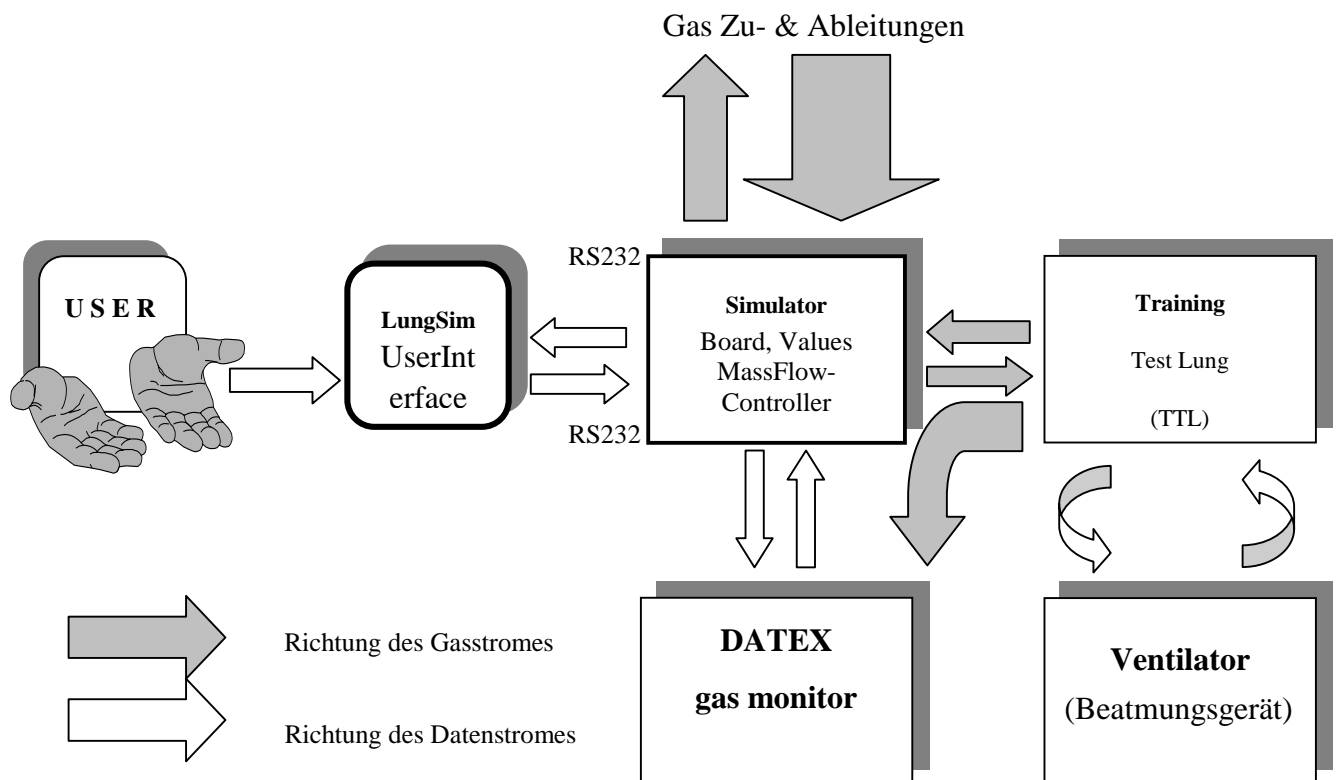


Bild 1 Schematische Aufbau

LungSim ist quasi das „Herzstück“ des Lungensimulators. Es steuert und überwacht alle am Projekt beteiligten Instrumente und Geräte. Der Benutzer kann nun durch seine Eingaben entscheiden, wie das Gesamtsystem sich verhalten soll.

5.3 Das Userinteface

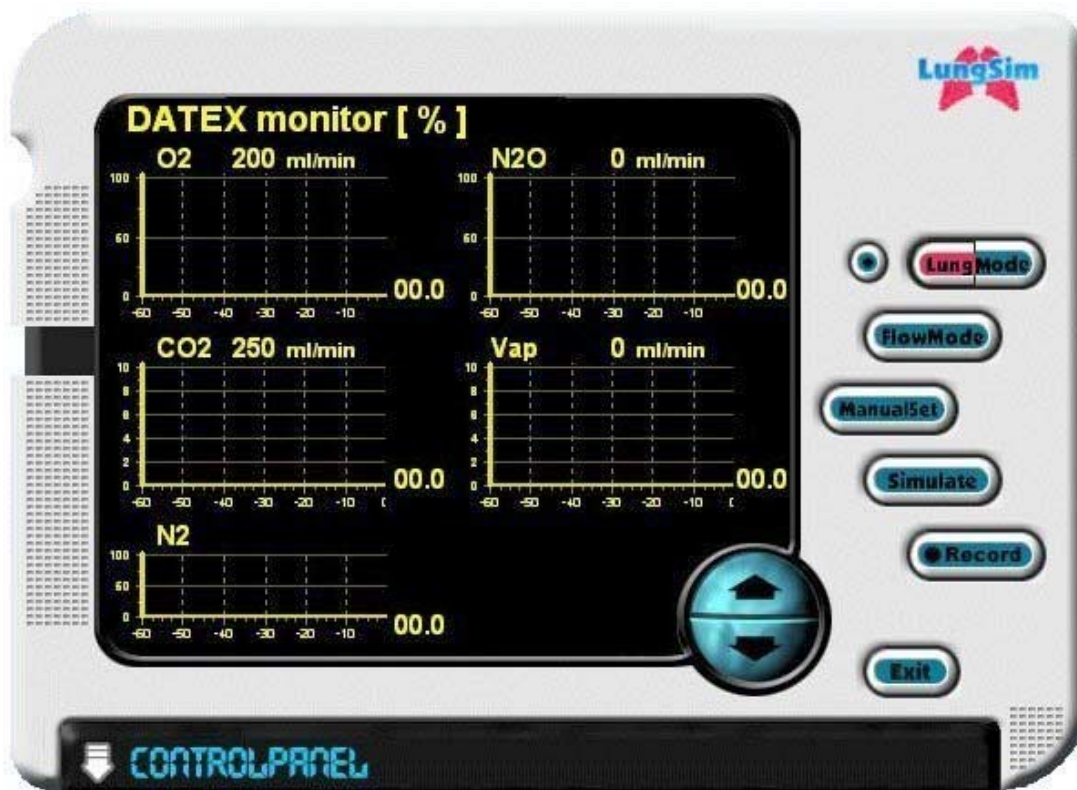


Bild 2 Screenshot „LungSim“ (verkleinerte Darstellung)

Die Optik der grafischen Benutzeroberfläche von *LungSim* wurde einem medizinischen Gerät nachempfunden. Das hat den schlichten Hintergrund, dass Mediziner bekanntlich im allgemeinen den Umgang mit „komplizierter“ Software scheuen. Dieser Scheu galt es nun durch ein ansprechendes Design, Übersichtlichkeit und selbsterklärenden Bedienkomfort entgegenzuwirken.

Ohne nun im einzelnen auf die medizinischen und funktionellen Aspekte der Software einzugehen, soll nun im folgendem die Bedienung von *LungSim* im Sinne der Softwareergonomie erläutert werden.

Beim Start des Programms erscheint eine Kurzanleitung auf dem Display, die dem Benutzer kurz und knapp die Ausführung der zur Verfügung stehenden Modi erläutert. Alle weiteren Eingaben oder Abläufe sind selbsterklärend bzw. intuitiv durchführbar. Stichwort *Selbstbeschreibungsfähig* (Punkt 4.3).

Geht man auf das Risiko ein, die Software *Selbstbeschreibungsfähig* zu programmieren, ergibt sich automatisch der unter Punkt 4.6 beschriebene Aspekt – die *Fehlerrobustheit*. Gesetzt den Fall, dass die Software für den Benutzer (weil er vielleicht nicht vom Fach ist) doch nicht selbsterklärend ist, darf sie zu keinem Zeitpunkt Eingaben annehmen, die ein Fehlverhalten auslösen oder falsche Ergebnisse liefern würde.

Nicht zu unterschätzen ist hierbei der enorm höhere Programmier- und Testaufwand, da quasi unzählig viele Kombinationen von Fehleingaben jeglicher Art zu jedem Zeitpunkt abgefangen werden müssen.

Bei LungSim erkennt der Benutzer selbst, ob die von ihm gerade gewünschte Funktion zur Verfügung steht.



Bild 3

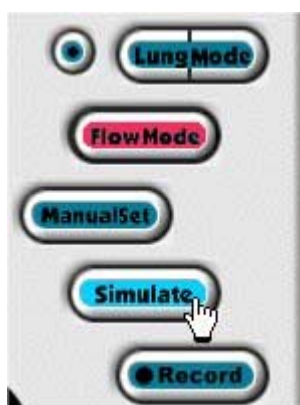


Bild 4

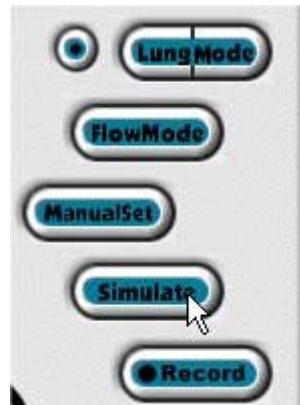


Bild 5

Berührt der Benutzer mit dem Mauszeiger einen Funktionsknopf, der zu diesem Zeitpunkt vom Programm zur Benutzung „freigegeben“ ist, so wird dies visuell durch ein Leuchten angezeigt (Bild 4) (manueller <-> visueller Dialog). Dieser Knopf kann gedrückt werden. Gedrückte Knöpfe erscheinen wie im Bild 3 zu sehen ist rot und nach innen leicht versenkt. Zum deaktivieren ist einfach noch einmal auf diesen Knopf zu klicken. Die Funktionsweise wurde einem Taster wie z.B. am einem Radio nachempfunden.

In der Gegenüberstellung von Bild 4 und 5 wird gezeigt, dass der Anwender in Bild 5 einen Fehler gemacht hat – „Simulate“ steht ihm nicht zur Verfügung. Bevor eine Simulation gestartet werden kann muss zunächst ein Modus ausgewählt sein. Erst dann kann der „Simulate“-Knopf betätigt werden.

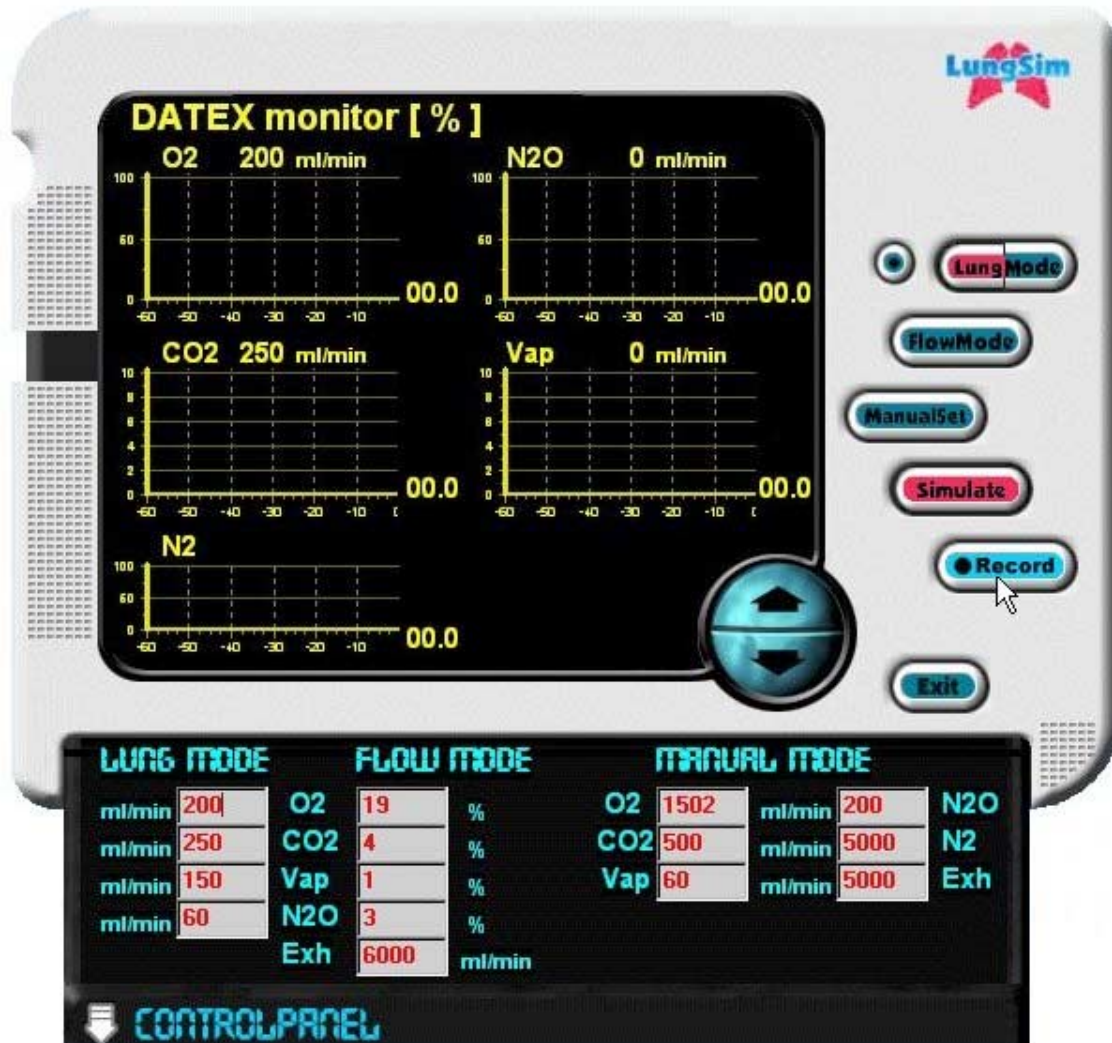


Bild 6 Screenshot „LungSim“ mit geöffnetem Panel (verkleinerte Darstellung)

Für die Eingabe von Daten steht dem Benutzer das ausfahrbare *controlpanel* zur Verfügung. In diesem Dialog stellen sich ihm die 3 verschiedenen Modi des Simulators dar. Mit Hilfe der Tastatur können hier die Werte „beliebig“ geändert werden. Auch hier musste auf *Fehlerrobustheit* geachtet werden, so dass die Werte eben doch nicht ganz „beliebig“ sein können. Wird z.B. an Stelle eines Kommas ein Punkt eingegeben, so wird dies automatisch korrigiert (Deutsches und Amerikanisches System sind uneinheitlich). Werden statt erwarteter Zahlen Buchstaben eingegeben ertönt ein kurzer Signalton und das Eingabefeld setzt automatisch den zu letzt genutzten „sinnvollen“ Wert ein. Das ist sehr wichtig, da die eingegebenen Daten während einer Simulation sofort in die Berechnungen mit eingehen.

Während der Simulation können alle auflaufenden Messdaten in Echtzeit auf dem Display dargestellt werden. Der Anwender kann entscheiden welche er davon sehen möchte und in welcher Form die Daten für ihn aufbereitet werden sollen.

Das Hauptaugenmerk bei der Entwicklung lag hierbei auf die Übersichtliche Gestaltung bei der Darstellung der Daten.

5.4 Schlussbemerkung

Um die *Selbstbeschreibungsfähigkeit*, *Benutzerzufriedenheit* und *Fehlerrobustheit* von *LungSim* zu verifizieren wurden während der Entwicklung und vor allem nach der „Fertigstellung“ etliche Tests gefahren. Ich habe *LungSim* von mehreren verschiedenen Personen bedienen und mir ihre Eindrücke, Kritiken und Anregungen schildern lassen. Wie ich festgestellt habe, ist diese Verfahrensweise eine sehr sehr gute Möglichkeit ein Feedback vom potentiellen Anwender zu erhalten. Nur so ist es möglich eine Software noch besser – noch ergonomischer zu machen.

6 Quellen und Literaturhinweise

Für diese Ausarbeitung wurde folgende Literatur zu gelesen und auch teilweise zitiert:

- „Softwareergonomie“ Techniken der Dialoggestaltung 2.Auflage von A.Zeidler und R.Zellner (R.Oldenbourg Verlag München Wien 1994)
- „Softwareergonomie“ Grundlagen der Mensch-Computer-Kommunikation 1.Auflage von Michael Herczeg (Addison-Wesley Verlag 1994)
- „Grundlagen der menschengerechten Arbeitsgestaltung“ Hans Martin (Bund – Verlag)
- Auszüge aus der DIN 66 234, Teil 8 und der ISO 9241