

## XIII VISIVAL

# Ein Baukasten für Bildschirm-Instrumente

Michael Herczeg

Die Arbeit mit interaktiven Computersystemen ist gekennzeichnet durch eine dichte, alternierende Abfolge von Eingaben des Benutzers und Ausgaben des Systems. Die Protokolle, d.h. die Regeln, die diesen Dialog leiten, sind im Computerprogramm festgeschrieben.

Die Erstellung eines interaktiven Anwendungssystems auf einem Computer setzt sich somit aus wenigstens zwei Teilaufgaben zusammen: zum einen aus der Bereitstellung der eigentlichen Anwendungsfunktionalität und zum anderen aus deren Anbindung an die Ein-/Ausgabegeräte. Insbesondere die Programmierung der Ein-/Ausgabe, man spricht vom Bau der Benutzerschnittstelle, läßt sich durch die Bereitstellung anwendungsneutraler und damit wiederverwendbarer Bausteine deutlich vereinfachen. Durch die Verwendung leistungsfähiger und individualisierbarer Bausteine lassen sich Benutzerschnittstellen schneller, konsistenter und trotzdem problemangemessener erstellen.

Am Beispiel des Baukastens VISIVAL mit Bausteinen zur vorwiegend graphischen Ein- und Ausgabe numerischer Daten soll diese Vorgehensweise verdeutlicht werden.

## 1. Einleitung

Typische Anwendungen, in denen die graphische Ein- und Ausgabe numerischer Werte eine problemangemessene Interaktionsform ist, sind vor allem die Überwachung und Steuerung realer oder simulierter Prozesse. Die Medien zur Ausgabe numerischer Werte nennt man in traditionellen - nicht zwangsläufig computergestützten - Anwendungen *Instrumente*. Die Darstellungen solcher Instrumente auf dem Computerbildschirm werden im folgenden ebenfalls Instrumente genannt. Diese können allerdings gegenüber ihren mechanischen „Vorbildern“ gleichzeitig zur Werteingabe und Wertausgabe benutzt werden.

Als Anwendungsbeispiel dient ein Monitor zur Überwachung wichtiger Zustandsparameter des Betriebssystems UNIX auf einem Rechner des Typs VAX 11/780. Beim VAX/UNIX-Monitor handelt es sich um ein reines Kontrollwerkzeug; die Eingabe von Werten wird daher nicht betrachtet werden.

Dieses Beispiel ist repräsentativ für eine große Klasse von Anwendungen. Überall dort, wo heute Menschen vor Instrumenten sitzen, um den Ablauf eines Prozesses zu kontrollieren, können sich in Zukunft Bildschirme mit Darstellungen von Instrumenten befinden, die genauer, flexibler, leichter modifizierbar, abnutzungsfrei und unzerstörbar die wichtigsten Kenndaten der Prozesse darstellen.

Als Prototyp eines Baukastens zur Erstellung von Bildschirminstrumenten dient das System VISIVAL<sup>1</sup>. Verwandte Arbeiten finden sich in den Systemen LOOPS [Stefik, Bobrow 86], KEE [KEE 85] und STEAMER [Hollan, Hutchins, Weitzmann 84].

## 2. Der VAX/UNIX-Monitor

Am Beispiel der Bedienoberfläche einiger Kommandos des UNIX-Betriebssystems soll gezeigt werden, wie ein Baukasten zum Bau von Benutzerschnittstellen gestaltet werden kann.

### 2.1 Das Problem aus Benutzersicht

Im UNIX-Betriebssystem gibt es eine Reihe von Dienstprogrammen, die einen Einblick in das Verhalten des Betriebssystems erlauben. Ein solcher Einblick ist notwendig, um das Betriebssystem möglichst gut an die ablaufenden Anwendungsprogramme anpassen zu können.

Durch das Dienstprogramm `vmstat` [UNIX 83] läßt sich beispielsweise das Paging-, Ein-/Ausgabe- und Time-Sharing-Verhalten sowie die Auslastung des Rechners kontrollieren (Abbildung 1). Das Dienstprogramm `ps` zeigt die Verteilung der Rechnerressourcen auf die Benutzer (Abbildung 2). Die Zahlenkolonnen der beiden Programme enthalten die zur Kontrolle von UNIX notwendigen Daten. Die Tabellen werden auf dem Bildschirm „gescrollt“, d.h. neue Tabellenzeilen schieben die vorausgehenden Darstellungen nach oben und lassen die Zeilen am oberen Bildschirmrand verschwinden.

---

<sup>1</sup> Das System VISIVAL wurde von H.-D. Stenger erstellt. Eine ausführliche Beschreibung findet sich in [Stenger 86]. Viele Bilder dieses Beitrags stammen aus dieser Arbeit.

Das Herausfiltern und Interpretieren dieser Daten fällt selbst Systemspezialisten schwer. Dies gilt insbesondere für das Abschätzen des zeitlichen Systemverhaltens sowie das Ergründen von Ursache und Wirkung. Gründe dafür sind die eigenwillige Codierung der Daten, die Einbettung wichtiger in unwichtige Daten, der Umfang der Daten und die aus diesen Bedingungen resultierende Überforderung des visuellen Aufnahmevermögens und Gedächtnisses des Betrachters.

```
csh(167): vmstat 2 20
```

procs		memory		re		at		pi	po	fr	page		disk		faults		cpu		id	
r	b	w	avm	fre	re	at	pi	po	fr	de	sr	ro	h0	h1	x3	in	sy	cs	us	id
1	0	0	1428	752	3	0	1	1	0	0	2	2	1	1	0	172	37	16	27	12
1	0	0	1480	709	0	0	1	0	0	0	0	0	0	3	0	231	215	17	84	13
1	0	0	1480	709	0	0	0	0	0	0	0	0	0	0	0	502	155	14	0	13
0	0	0	1343	724	0	0	0	0	0	0	0	0	0	0	0	351	105	11	37	5
0	0	0	1343	724	0	0	0	0	0	0	0	2	0	0	0	264	77	12	13	6
0	0	0	1343	724	0	0	0	0	0	0	0	0	1	0	0	212	55	9	22	8
2	0	0	1368	730	0	0	0	0	0	0	0	0	2	3	0	164	83	16	61	36
2	0	0	1368	730	0	3	3	0	0	0	0	0	3	14	0	313	74	25	17	33
0	0	0	1321	706	0	1	2	0	0	0	0	0	0	0	0	253	64	18	14	5
0	0	0	1321	706	0	0	2	0	0	0	0	2	4	0	0	203	65	16	10	20
1	0	0	1423	716	4	0	3	0	0	0	0	1	3	0	0	160	64	15	4	13
1	0	0	1423	716	2	0	5	0	0	0	0	7	11	1	0	127	102	25	12	45
1	0	0	1423	716	5	0	9	0	0	0	0	14	7	17	0	123	111	31	8	44
1	1	0	1744	662	3	0	8	0	0	0	0	5	13	4	0	169	148	41	43	45
1	1	0	1744	662	1	0	20	0	0	0	0	0	49	0	0	152	114	59	10	30
1	0	0	2156	551	0	0	30	0	0	0	3	1	47	0	0	146	79	75	17	28
1	0	0	2156	551	0	0	26	15	12	0	36	5	19	3	0	125	57	68	36	17
1	0	0	2156	551	0	0	18	13	14	0	32	0	3	6	0	102	44	55	73	14
1	0	0	2152	448	0	0	12	8	10	0	21	0	3	0	0	82	32	41	91	6
1	0	0	2152	448	0	0	7	5	6	0	12	0	0	0	0	66	23	30	98	2

Abbildung 1. Das UNIX-Dienstprogramm vmstat.

```
csh(158): ps -au
```

USER	PID	%CPU	%MEM	SZ	RSS	TT	STAT	TIME	COMMAND
michael	26060	44.4	2.0	170	139	01	R	0:04	bgx -2 vmstat.bgs
michael	26061	36.8	1.4	158	95	01	R	0:01	ps -au
joachim	24139	23.9	4.9	665	344	10	R	5:11	emacs
juergen	25800	3.0	50.9	5763	3623	05	S	4:03	elab
joachim	25771	1.9	3.3	443	227	15	I	1:02	emacs test.l
martin	26048	0.2	0.6	114	37	01	I	0:03	-csh (csh)
root	26047	0.1	0.3	57	15	01	I	0:00	/etc/telnetd
juergen	25913	0.0	3.7	427	256	05	T	0:19	emacs ablauf.mss
michael	26003	0.0	3.7	443	256	01	T	0:06	emacs visival.mss
notes	24918	0.0	0.6	152	34	04	I	0:15	notes
michael	25890	0.0	1.2	169	77	01	T	1:08	scbg master
joachim	25752	0.0	5.0	3809	325	15	T	0:17	bigw1isp
joachim	24158	0.0	4.1	1320	284	10	T	0:37	/inform/cmd/post
dorle	23665	0.0	0.0	25	0	22	TW	0:26	emacs
dorle	24141	0.0	0.0	4462	0	22	TW	2:45	snap
martin	23091	0.0	0.0	1336	52	20	TW	0:33	post
kalle	24241	0.0	0.0	5432	0	08	IW	7:39	/inform/demos/help/demo
dorle	23886	0.0	0.0	595	31	08	IW	2:22	emacs
dorle	25742	0.0	0.0	169	28	22	IW	0:26	scbg otello
martin	25082	0.0	0.0	101	9	20	IW	0:00	/bin/csh /inform/cmd/toff
dorle	23666	0.0	0.0	542	31	22	IW	1:45	emacs
klaus	25728	0.0	0.0	101	9	11	IW	0:00	/bin/csh /inform/cmd/toff
joachim	25562	0.0	0.0	122	9	00	IW	0:10	-csh (csh)
root	25561	0.0	0.2	57	8	00	I	0:00	/etc/telnetd

Abbildung 2. Das UNIX-Dienstprogramm ps.

In Abbildung 3 sieht man eine Alternative, den VAX/UNIX-Monitor, der neben anderen die gleichen Systemparameter wie `vmstat` und `ps` visualisiert. Die Daten werden durch Instrumente übersichtlich dargestellt, zeitliche Verläufe werden „auf einen Blick“ sichtbar. Die hohe Leistung des menschlichen visuellen Wahrnehmungsvermögens wird dabei ausgenutzt. Relative Größen können direkt erkannt werden und müssen nicht rechnerisch erarbeitet werden. Dort, wo exakte Werte von Bedeutung sein können, werden sie in den Instrumenten zusätzlich zur graphischen Darstellung digital angezeigt. Die Anzeigen in den Instrumenten verändern sich dynamisch bei Veränderung der überwachten Größen.

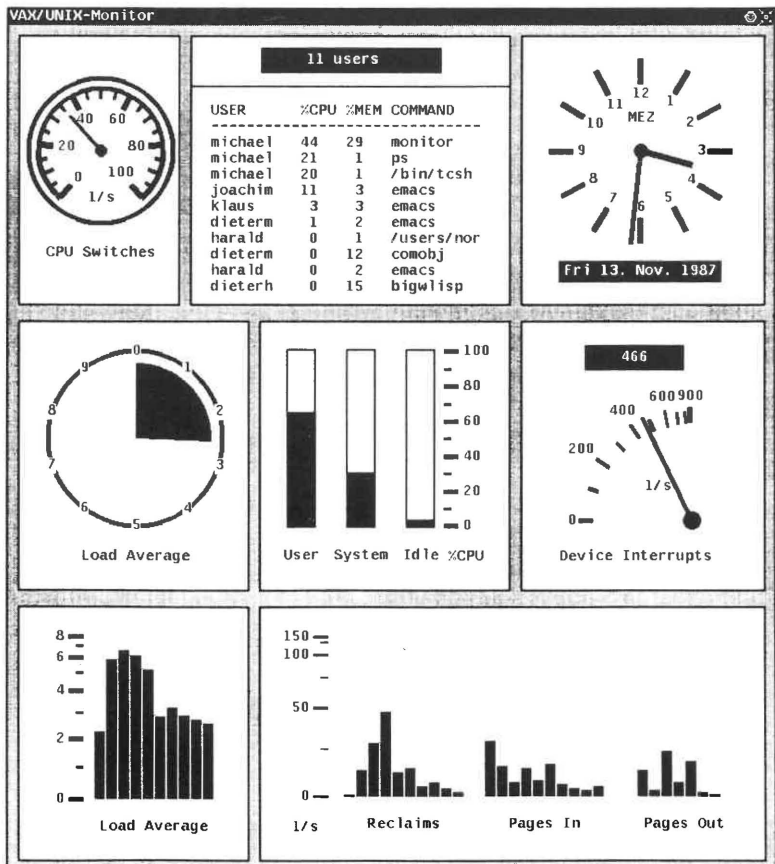


Abbildung 3. Der VAX/UNIX-Monitor.

## 2.2 Grundtypen von Instrumenten

Im VAX/UNIX-Monitor werden verschiedenartige Instrumente eingesetzt. Sie können nach Darstellungsform, Anzahl der dargestellten Werte und der Sichtbarkeit der zeitlichen Werteentwicklung (Wertehistorie) unterschieden werden.

### Darstellungsform

Je nach notwendiger Genauigkeit oder Übersichtlichkeit der dargestellten Zahlenwerte lassen sich *digitale* und *analoge Instrumente* einsetzen. Digitale Instrumente stellen die Werte in Zifferschreibweise mit beliebiger Genauigkeit dar; Analoginstrumente zeigen den Zahlenwert graphisch durch einen Zeiger auf einer Skala an. Zeiger und Skala können dabei reichhaltig variiert und damit an die Anwendung und den Benutzer angepaßt werden. Wenn nötig, können Digital- und Analoginstrumente zu einem kombinierten Instrument zusammengesetzt werden (Abbildung 4).

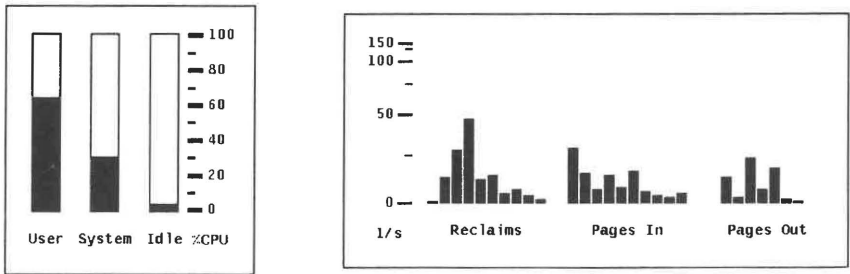


Abbildung 4. Analoges, digitales und kombiniertes Instrument.

### Anzahl der dargestellten Werte

Innerhalb eines Instruments können auch mehrere Zahlenwerte dargestellt werden. Dies bietet sich zum Beispiel bei der Darstellung logisch zusammengehöriger Werte oder bei Darstellung der zeitlichen Entwicklung eines Zahlenwerts an (Abbildung 5).

Abbildung 5 zeigt links ein Instrument mit der Aufteilung der CPU-Leistung auf Anteile für die Benutzer, für das System und, als eventuell verbleibender Rest, die nicht ausgenutzte Leistung. Die Werte sind jeweils auf 100% bezogen und bilden damit relative Wertdarstellungen, die leicht überblickt werden können. Die Skala spielt hier nur eine untergeordnete Rolle.



**Abbildung 5.** Ein Instrument mit drei Zahlenwerten und eines mit drei Werthistorien.

## Werthistorien

Rechts in Abbildung 5 sieht man die zeitliche Entwicklung des Pagingverhaltens der Maschine. Jeder neue Wert wird als schwarzer Balken rechts angehängt. Wird die darzustellende Anzahl der Zahlenwerte (hier zehn) überschritten, so wird der älteste Wert entfernt und die Darstellung nach links verschoben. Die Darstellung der Einzelwerte kann durch verschiedene Zeiger erfolgen; in diesem Beispiel wurden wertproportionale Flächen in Form von Balken gewählt.

## Manipulierbarkeit visualisierter Werte

Aus der Erfahrung kennt man vor allem Anzeigeeinstrumente. Es kann jedoch auch sinnvoll sein, die angezeigten Werte vom Benutzer ändern zu lassen. Bei digitalen Instrumenten kann dies durch Eintippen des gewünschten Werts mittels Tastatur, bei analogen Instrumenten durch Verstellen des Zeigers oder durch Deuten auf einen Zielwert der Skala erfolgen. Dies ist ein wichtiger Vorteil von Bildschirminstrumenten gegenüber mechanischen Instrumenten. Im VAX/UNIX-Monitor könnte man verschiedene Betriebssystemparameter zum Verstellen anbieten, z.B. die maximal zulässige Anzahl von Benutzern im System.

## 2.3 Bestandteile der Instrumente

Instrumente lassen sich sowohl konzeptionell als auch implementierungstechnisch in verschiedene Bestandteile zergliedern. In erster Linie sind dies Anzeigen, Skalen, Beschriftungen, Alarmer sowie Über- und Unterlaufanzeigen.

## Anzeigen

Eine *Anzeige* ist die eigentliche Visualisierung eines Werts. In digitalen Instrumenten ist dies die dargestellte Zahl, in analogen Instrumenten der Zeiger.

Digitale Anzeigen können durch verschiedene Zeichensätze (*Fonts*) in Größe und Schriftart variiert werden. Die Zahl kann u.a. als Festpunktzahl, Gleitpunktzahl oder Exponentialdarstellung ausgegeben werden. Sie kann zentriert, rechts- oder linksbündig positioniert werden. So lassen sich diese Instrumente an die von der Anwendung abhängigen Wertebereiche anpassen.

Auch analoge Anzeigen können vielfältig variiert werden. Wertproportionale Flächen oder Zeiger visualisieren den aktuellen Wert, meist in Bezug auf eine Skala. Typische Zeiger bewegen sich entweder kreisförmig oder linear horizontal oder linear vertikal (Abbildung 6). Die Zeigerform kann durch Rechtecke, Dreiecke, Kreise oder durch Kombinationen davon beschrieben werden.

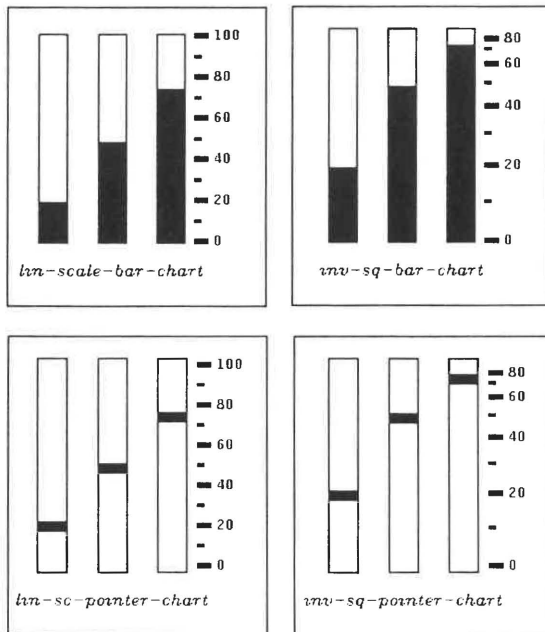
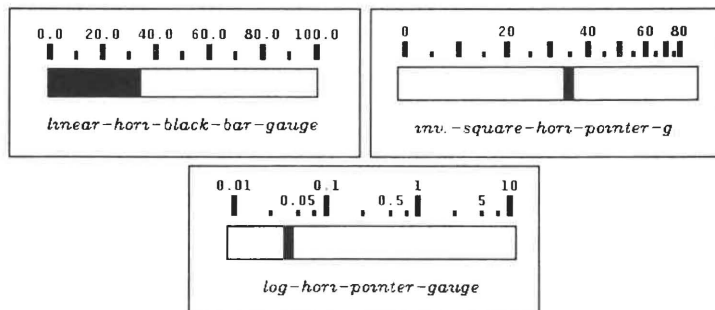


Abbildung 6. Variation von Zeigern und Skalen in linearen Instrumenten.

## Skalen

Analoge Anzeigen sind meist mit einer oder mehreren *Skalen* gekoppelt. Nur so kann aus der Position eines Zeigers der Anzeigewert abgeleitet werden. Entsprechend den Instrumenten sind Skalen kreisförmig oder linear angeordnet. Sie enthalten Markierungen und Beschriftungen, an denen die aktuellen Werte abgelesen werden können. Die dargestellten Wertebereiche sind entweder linear gestaffelt oder über eine mathematische Funktion (häufig Logarithmus oder Quadrat) verzerrt (Abbildung 7). Damit kann der Wichtigkeit verschiedener Bereiche des dargestellten Wertebereichs Rechnung getragen werden. Im *Instrument Device Interrupts* sowie in den beiden untersten Instrumenten (*Load-Average* und *Paging*) im *VAX/UNIX-Monitor* werden nichtlineare Skalen verwendet (Abbildung 3).



**Abbildung 7.** Verschiedene Skalen in horizontal linearen Instrumenten.

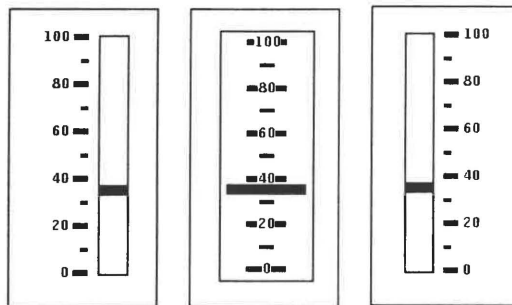
## Beschriftungen

Die Bedeutung eines Instruments bzw. seiner Anzeigen wird durch *Beschriftungen* gekennzeichnet, die durch vielerlei Zeichensätze in Größe und Schriftart variiert werden können. Auch die Position oder Anordnung einer Beschriftung innerhalb des Instruments ist für seine Nutzung von Bedeutung. Für die unterschiedlichen Instrumententypen gibt es jeweils standardisierte günstige Anordnungen für Beschriftungen (Abbildung 8).

## Alarme

Inbesondere bei Prozeßüberwachungen haben dargestellte Größen meist Grenzwerte, deren Unter- oder Überschreitung vermieden werden soll. Tritt eine solche Größe aus dem zulässigen Bereich heraus, soll ein *Alarm* erfol-





**Abbildung 8.** Standardanordnungen für Beschriftungen eines linearen Instruments.

gen, der aus einem akustischen bzw. einem optischen Signal bestehen kann. Für den Fall, daß der Beobachter bei Auftreten des Alarms das Instrument gerade nicht beobachtet, sollte eine deutlich sichtbare zusätzliche Markierung des Instruments vorgesehen werden, die erst durch Bestätigung der überwachenden Person beseitigt wird. Im VAX/UNIX-Monitor tritt ein solcher Alarm beispielsweise bei einem **Load Average** von mehr als acht ein. In diesem Fall sollte der Operateur eingreifen und unwichtige Prozesse in der Priorität vermindern oder ganz anhalten.

Bei einem Alarm kann zusätzlich eine beliebige Aktion im Anwendungssystem ausgelöst werden. Instrumente können auf diese Weise nicht nur zur Überwachung, sondern auch zur Prozeßsteuerung eingesetzt werden. Es ist dann auch zwischen dem Unterschreiten eines Minimalwerts und dem Überschreiten eines Maximalwerts zu unterscheiden, da dies im allgemeinen unterschiedlicher Reaktionen bedarf.

## Über- und Unterlaufanzeigen

Gelegentlich reicht die in einem analogen Instrument vorhandene Skala nicht mehr aus, um einen Wert anzuzeigen. In solchen Fällen kann ein zusätzliches digitales Instrument sichtbar gemacht werden, um den Wert darzustellen (Abbildung 9): aus analogen Instrumenten werden damit kombinierte Instrumente. Solange sich der darzustellende Zahlenwert innerhalb des vorhandenen Skalenbereichs befindet, kann dieses *Über-/Unterlaufinstrument* unsichtbar bleiben. Tritt der Wert jedoch aus dem darstellbaren Bereich, wird es sichtbar und überlagert das Analoginstrument.

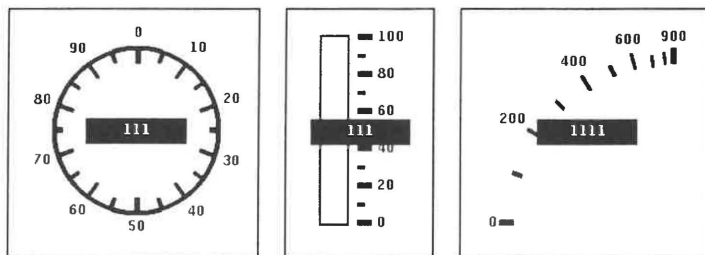


Abbildung 9. Instrumente bei Überlauf.

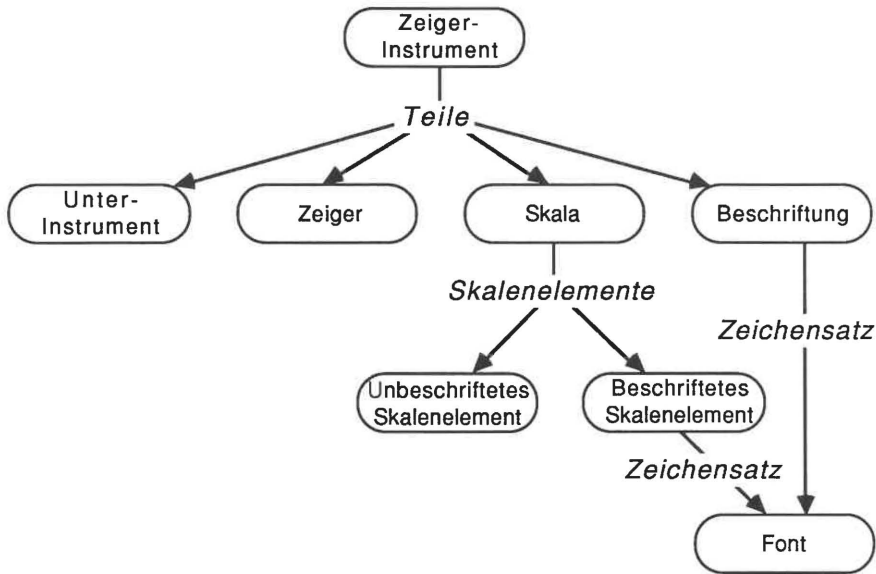
Durch diese Technik kann die Skala eines Instruments auf den Normalbetrieb ausgelegt werden, damit die Werte gut abzulesen sind. Wird dann der Maximalwert gelegentlich über- oder unterschritten, geht das Instrument nicht wie bei mechanischen Instrumenten in einen nutzlosen Endausschlag, sondern zeigt den Wert digital an. Alternativ wäre ein dynamisches Verändern des Skalenbereichs möglich; dies hat jedoch den Nachteil, daß sich der Beobachter nicht ohne Ablesen des genauen Werts orientieren kann, da sich die Skalen ständig beliebig verändern könnten. Bei einer routinemäßigen Überwachung wäre dies wegen möglicher Fehlinterpretation einer dargestellten Größe problematisch.

### 3. Die Systemstruktur von VISIVAL

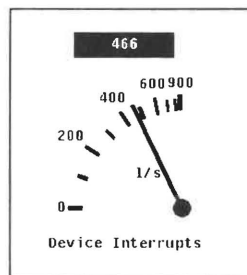
Der logische Aufbau und der Variationsreichtum von Instrumenten fordern geradezu, Instrumente mittels eines Baukastens konfigurieren zu können. Die Bausteine sind die Anzeigen, Skalen, Beschriftungen und Alarme. Darüber hinaus können zum Beispiel Über-/Unterlaufanzeigen und andere Instrumente (Abbildung 10) in ein Instrument eingebaut werden. Man erhält auf diese Weise hierarchisch gegliederte komplexe Instrumente.

Aufgrund des Gegenstandscharakters dieser Bausteine bietet es sich an, einen solchen Baukasten sowohl konzeptionell als auch programmtechnisch objektorientiert zu gestalten. Ein Alarmobjekt hat beispielsweise die Attribute *unterer Grenzwert* und *oberer Grenzwert* sowie die Methoden (d.h. funktionale Elemente) *Ausgabe der Warnung* (z.B. Warnton), *Aktion bei Unterlauf* und *Aktion bei Überlauf*. Die Attribute und Methoden können für die jeweiligen Anwendungen verschieden definiert werden.

Die verschiedenartigen Bausteine, die zu Instrumenten gekoppelt werden, besitzen jeweils eine Vielzahl von Attributen und Methoden, die in Objekt-



**Abbildung 10.** Aufbau des Instruments aus Abbildung 11.



**Abbildung 11.** Instrument, das sich gemäß Abbildung 10 zusammensetzt.

klassen beschrieben werden. Nun weisen selbst die Bausteine von Instrumenten wieder vielfältige Differenzierungen auf, die objektorientiert durch Klassenhierarchien beschrieben werden. Für die verschiedenen Zeiger ergibt sich die Klassenhierarchie in Abbildung 12.

Die Erstellung eines Instruments für eine bestimmte Anwendung besteht aus der Auswahl, Spezifikation und Generierung der Bestandteile des Instruments aus den vorhandenen Klassen. Dieser Konfigurationsprozeß ist selbst bei guter Dokumentation eines Baukastens, der beispielsweise wie VISIVAL

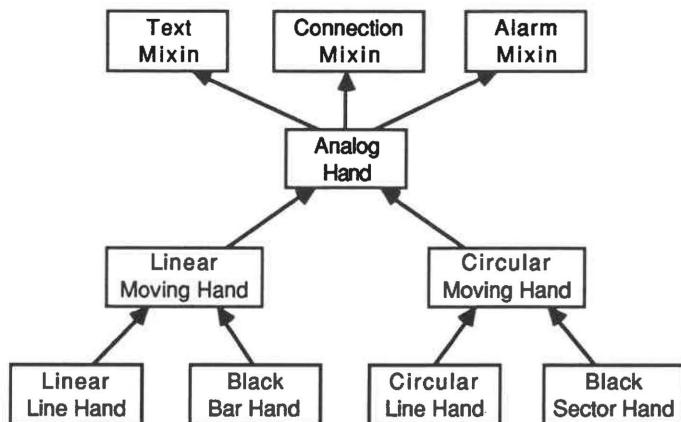
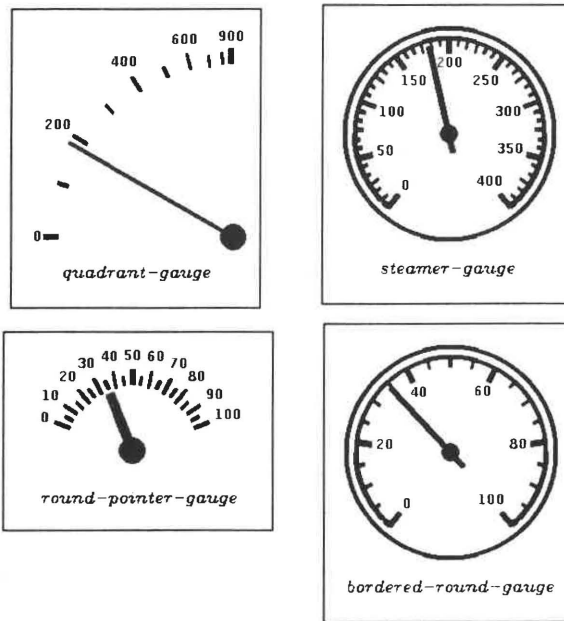


Abbildung 12. Klassenhierarchie von Zeigern.

mehr als 200 Bausteine in Form von Klassen besitzt, ein nicht mehr ganz einfaches und daher fehleranfälliges Verfahren. Für das gewünschte Aussehen und Verhalten eines Instruments müssen die richtigen Bausteine erzeugt und gekoppelt werden. Nicht alle Kombinationen von Bausteinen sind sinnvoll und verträglich. So macht es beispielsweise wenig Sinn, eine kreisförmige Skala mit einem linear bewegten Zeiger zu verbinden. Es gibt eine ganze Reihe solcher Restriktionen bezüglich der Kombinierbarkeit von Bausteinen sowie deren Attributierung. Das vorhandene Spektrum der Darstellungsmöglichkeiten erlaubt aber das Maßschneidern von Instrumenten für Benutzer und Anwendung. In Abbildung 13 finden sich exemplarisch die Variationen runder Instrumente.

Benutzerschnittstellen-Baukästen sollen den Anwendungsprogrammierer so weit wie möglich entlasten. Daher bietet es sich an, die Konfiguration von Instrumenten durch ein *interaktives Konstruktorsystem* zu unterstützen. Der Anwendungsprogrammierer spezifiziert damit - auf Wunsch systemgeführt - wie die Instrumente aussehen sollen. Das ganze Spektrum an Möglichkeiten wird dabei vor ihm ausgebreitet. Seine Entscheidungen werden durch die Darstellung eines Beispielinstruments verdeutlicht. Vom gesamten Designraum werden ihm die jeweils noch verbleibenden sinnvollen Freiheitsgrade angeboten und eventuell an Beispielen erläutert. Das Konstruktorsystem benötigt dazu u.a. Wissen über Verträglichkeiten bzw. Unverträglichkeiten zwischen den einzelnen Bausteinen.



**Abbildung 13.** Verschiedene kreisförmige Instrumente.

Ist ein Instrument fertiggestellt, kann es vom Konstruktorsystem an das Anwendungssystem angekoppelt werden. Ist dieses noch nicht vorhanden, können Ein- und Ausgaben simuliert und so das spätere Verhalten des Instruments getestet werden.

Ein solches Konstruktorsystem hat sich bei allen komplexen Baukästen als wichtige und oft notwendige Unterstützung herausgestellt. Für VISIVAL ist ein solches in Planung.

#### 4. Zusammenfassung und Ausblick

Instrumente, wie sie beispielsweise mit VISIVAL erzeugt werden können, sind nur eine Komponente eines Benutzerschnittstellen-Baukastens. Weitere Komponenten sind beispielsweise Formulare, Tabellen, Icons und Menüs. Jedes dieser Interaktionsmedien läßt sich wiederum durch einen eigenen Baukasten modular aufbauen und konfigurieren, wobei diese Baukästen dann in allgemeineren Baukastensystemen eine gemeinsame Basis besitzen können. Man gelangt so zu einem komplexen System aus ineinanderge-

schachtelten, aufeinanderaufbauenden oder nebeneinanderstehenden Modulen, die als Gesamtheit den Kern eines sogenannten *User Interface Management Systems (UIMS)* bilden. Konstruktorsysteme, Anwendungssimulatoren und Benutzermodelle vervollständigen dieses UIMS weiter. Sowohl ergonomischen als auch wirtschaftlichen Forderungen kommt man mit solchen Systemen wesentlich entgegen [Herczeg 86a].

Wenn bislang vorwiegend Büroarbeiten durch moderne Benutzerschnittstellen mit Icons, Menüs und Formularen unterstützt werden konnten, so werden durch Systeme wie VISIVAL in Zukunft auch Techniker ihre Arbeitswelt auf dem Bildschirm wiederfinden. In vielen Fällen werden sie ihre Anwendungen selbst mit geeigneten Interaktionsmedien (wie z.B. Instrumenten) versorgen; ein Beispiel dafür ist das computergestützte Elektronik-Simulationslabor ELAB, das in Kapitel II beschrieben ist.

Als programmtechnische Basis für derartige Benutzerschnittstellen-Baukästen haben sich objekt- und frameorientierte Sprachen als sehr geeignet herausgestellt. Bei VISIVAL ist es die Sprache ObjTalk [Rathke C. 86b]. Als Ausgabegerät wird ein hochauflösender Rasterbildschirm verwendet, als Zeigeinstrument eine Maus.

Zu erwartende Verbesserungen in der Geschwindigkeit und der Feinheit der graphischen Ausgabe auf dem Bildschirm werden es ermöglichen, daß Bildschirminstrumente realen physikalischen Instrumenten in der Darstellungsqualität praktisch nicht mehr nachstehen. In Genauigkeit, Flexibilität und Robustheit sind sie ihnen heute schon überlegen. Der Trend zu Bildschirminstrumenten zeichnet sich bei aufwendigen technischen Überwachungssystemen wie zum Beispiel Flugzeugcockpits schon heute ab.