

COMPUTERUNTERSTÜTZTE AUSLEGUNG VON FLUGZEUGSYSTEMEN AM BEISPIEL DER FLUGSTEUERUNG

D. Scholz

Arbeitsbereich Flugzeug-Systemtechnik
Technische Universität Hamburg-Harburg

1. ZUSAMMENFASSUNG

Flugzeugsysteme - auch bekannt als Grund- oder Ausrüstungssysteme - haben in modernen Transportflugzeugen einen hohen technischen und wirtschaftlichen Stellenwert erreicht. Für den Vorentwurf der immer komplexer werdenden Flugzeugsysteme sucht man spezielle Computerprogramme jedoch bisher vergebens. Es besteht ein Bedarf für ein Computer Aided Engineering (CAE) der Flugzeugsysteme und damit für ein geeignetes DV-Werkzeug.

Flugzeugsysteme fliegen erfolgreich in modernen Flugzeugen. Bei näherer Betrachtung kann jedoch festgestellt werden, daß Standard-Entwurfsverfahren für Flugzeugsysteme oft nicht verfügbar sind. Der Programmierung eines CAE-Tools für den Vorentwurf von Flugzeugsystemen muß daher eine Aufarbeitung der Theorie vorangestellt werden.

Bei der Programmierung eines CAE-Tools sind die Möglichkeiten auszunutzen, die durch die heutige Computertechnik gegeben sind. Graphische Ein- und Ausgabe, Benutzerfreundlichkeit und die Beachtung von Oberflächenstandards sind heute Voraussetzung für die Akzeptanz durch den Anwender. Dies erfordert eine hohe Investition an Programmieraufwand. Durch die konsequente Beachtung einer hardwareunabhängigen Programmierung - besonders bei der Erstellung der Gaphik - kann diese Investition geschützt werden.

2. EINLEITUNG

Die Ingenieurarbeit während Vorentwicklung und Entwicklung besteht bekanntlich darin, die Details des Entwurfes schriftlich festzuhalten. Beim Bau von zivilen Transportflugzeugen werden mit fortschreitendem Gang der Entwicklung u.a. Dokumente erstellt wie in Bild 1 dargestellt. Computer werden zur Unterstützung herangezogen. Zur Erstellung der *Aircraft Definition Note* eignen sich Flugzeugentwurfsprogramme wie z.B. [1] und [2]. Zur Erstellung einer *System- oder Sub-System-Specification* werden CAD, Simulationsprogramme oder Spezialprogramme der Fachabteilungen herangezogen. Für die Vorentwicklung von Flugzeugsystemen (die Vorentwicklung schließt etwa mit der Erstellung der *System Definition Note* ab), fehlt derzeit eine gezielte Computerunterstützung.

Diese Lücke kann mit einem DV-Werkzeug geschlossen werden, welches sich anlehnt an den Aufbau und das Vorgehen, wie er bei Flugzeugentwurfsprogrammen bekannt ist (vergleiche u.a. [3]). Dabei sollte auf Daten aufgebaut werden, wie sie in der *Aircraft Definition Note* enthalten sind.

In diesem Beitrag soll das Konzept des CAE-Tools vorgestellt werden. Anschließend soll dann auf die Grundlagen der Programmierung eingegangen werden. Mathematische Grundlagen zum Tool werden in diesem Beitrag nicht behandelt. Diese wurden bisher in [4], [5] und [6] dargestellt.

3. PROGRAMM-MODULE

Die Module des DV-Werkzeuges zum Entwurf von Flugsteuerungs- und Hydrauliksystemen sind in Bild 2 gezeigt. Aus den Daten der *Aircraft Definition Note* werden solche aerodynamischen Derivativa berechnet, wie sie für nachfolgende Rechnungen benötigt werden. Die statische und dynamische Auslegung der Flugsteuerungsaktuatoren ist möglich, wenn Ruderscharniermomente und Stellgeschwindigkeiten unter Beachtung des verfügbaren Einbauraums und der allgemeinen Spezifikationsdaten für Stellsysteme bekannt sind. Die Flugsteuerungskonfiguration - also die Zuordnung von Energie- und Signalquellen auf die Flugsteuerungsaktuatoren - wird so festgelegt, daß die höchste Manövrierbarkeit des Flugzeugs unter Normal- und Ausfallzuständen erreicht wird. Mit der Flugsteuerungskonfiguration, sowie den Anforderungen der Flugsteuerung und anderer Verbraucher an das Hydrauliksystem kann das Hydrauliksystem ausgelegt werden. Mit einer Entwurfsbewertung durch eine speziell an Flugzeugsysteme angepaßte DOC-Methode kann der Entwurf bewertet werden.

Die Entwurfsunterstützung durch das CAE-Tool ist rein algorithmisch. Denkbar wäre auch der Einsatz eines Expertensystems wie Bild 3 am Beispiel eines Entwurfes der Firma *Rockwell International* zeigt [7].

Die Programm-Module des CAE-Tools werden auf einer graphischen Oberfläche über Icons gestartet. Bild 4 zeigt dies am Beispiel einer Windows-Oberfläche. Über weitere Icon kann zu jedem Modul jeweils ein Hypertext-System aktiviert werden. Das Hypertext-System enthält eine "Wissensbasis" mit technischen Hintergründen zum Pro-

gramm-Modul, Lösungsstrategien und "Lessons Learned". Dienstprogramme z.B. für die graphische Darstellung von Berechnungsergebnissen stehen zur Verfügung.

Die Module bestehen aus einem Hauptfenster, welches die Eingabewerte aufnimmt (Bild 5). Nach dem Start der Berechnung werden die Ergebnisse in einem Ergebnisfenster angezeigt. Eine graphische Aufbereitung der Ergebnisse ist mit Hilfe eines technisch-wissenschaftlichen Plot-Programms (GNUPLOT, ein Public Domain Programm) möglich, welches in die Oberfläche integriert wurde. Das Hypertext-System zum jeweiligen Programm-Modul kann ebenfalls über das "Hilfe-Menü" gestartet werden.

Die Daten werden automatisch in einer universellen Datenbasis gespeichert und aus dieser bei Bedarf auch wieder ausgelesen. Die Datenbasis ist eine ASCII-Datei, die bei Bedarf von anderen Programmen (Editor, Datenbank, Tabellenkalkulationsprogramm, ...) bearbeitet werden kann. Die einzelnen Daten der Datenbasis werden durch ein Schlüsselwort identifiziert. Das Schlüsselwort wird aus verschiedenen Teilen gebildet. Diese Teile des Schlüsselwortes gehen aus einer hierarchischen Gliederung möglicher Daten hervor.

Die Module sind selbständig lauffähig. Zusätzliche Software ist nicht erforderlich.

Die Softwarequalität wird durch eine konsequente Programmierung nach einer eigens erstellten Programmierrichtlinie sichergestellt. Zusätzlich werden die einschlägigen DIN und ISO-Normen (ISO 9241, DIN 66234) bei der Erstellung der graphischen Oberfläche beachtet.

Die Bilder 6 bis 8 zeigen "Momentaufnahmen" weiterer Module. Skizzen (Bild 6) definieren die Eingabewerte. Ergebnisse werden - wann immer dies sinnvoll ist - graphisch dargestellt, wie in Bild 7 z.B. die Lage des Aktuators im Einbauraum oder wie in Bild 8 der Wert des effektiven Hebelarms eines Aktuators als Funktion des Ruderausschlags.

Ergebnisse, die anderen Modulen wieder als Eingabewerte dienen, können in der Datenbasis gespeichert werden. In dieser Weise werden Daten von einem Modul an das nächste weiter gegeben.

4. PROGRAMMIERUNG

Früher bestand eine Applikation aus einem mehr oder weniger homogenen Gebilde von Programmzeilen, die sowohl die Bildschirmausgabe als auch die Verbindung zur Datenbasis übernahmen. Heute sollte die Applikation nur noch aus dem Programmcode bestehen, der die eigentliche Rechnung ausführt (Bild 9). Programmcode zur Darstellung der Ein- und Ausgabe am Bildschirm sollte von der Applikation getrennt sein. Zur Erledigung dieser Aufgabe bietet sich ein User Interface Management System (UIMS) an. Über das UIMS wird das Aussehen der Oberfläche einschließlich der Reaktion

auf beliebige Eingaben definiert. Ein Datenbank Management System (DBMS) übernimmt die Verbindung zur Datenbasis. Das DBMS wurde für die sehr einfache Datenverwaltung beim CAE-Tool selbst programmiert. Die Funktionalität des DBMS wurde oben bereits beschrieben.

Die durch das User Interface Management Systems (UIMS) zu lösende Aufgabe ist nach dem *Seeheim-Modell* [8] (Bild 10) strukturiert. Das *Seeheim-Modell* ist ein Schichtenmodell, welches eine Trennung von Präsentationsschicht und Anwendungsschicht vorsieht. Die Verbindung wird über "Messages" hergestellt, die über die "Dialogkontrolle" gesteuert werden. In Ausnahmefällen ist auch ein direkter Zugriff von der Anwendung auf die Präsentationsebene möglich.

Die zur Programmierung des CAE-Tools erstellte Umgebung ist in Bild 11 gezeigt. Der PC unter DOS und Microsoft Windows wurde als primäre Entwicklungsplattform gewählt, weil hier (gegenüber Workstations) die größten Restriktionen bei der Programmierung zu beachten sind und somit eine Portierung auf andere Systeme möglich wird. Die Applikation wird mit einem C-Compiler (MS C/C++ 7.0) erstellt. Um die Portabilität zu sichern, wird nur der ANSI-Sprachumfang genutzt. Es hat sich gezeigt, daß der so erstellte C-Quellcode nahezu ohne Modifikationen auf eine Workstation kompiliert werden kann.

Die Graphik wird mit OpenUI, einem UIMS der Firma *Open Software Associates* erstellt. *OpenUI* ermöglicht eine hardwareunabhängige Programmierung der Graphik. Bilder, die mit dem Zeichenprogramm *CorelDraw* erstellt wurden, können mit Hilfe von *OpenUI* eingebunden werden. Die Oberflächeninformationen werden in einer Datei abgespeichert, die als "Dictionary" bezeichnet wird. Das Dictionary muß zur Laufzeit der Programme neben der ausführbaren Datei (*.exe) auf dem Rechner verfügbar sein. Die Anpassung an die spezielle Hardware geschieht durch die "OpenUI Library", die direkt auf die Schnittstellen der Fenstersysteme (Windows, Motif, ...) zugreift. Über "Context Identifier" wird die Programmierung eines kontextsensitiven Zugangs zum Hypertextsystem möglich. *OpenUI* sorgt für das richtige Zusammenspiel zwischen Oberfläche und Hypertextsystem.

Das Hypertextsystem wird über die Windows Hilfe als primäre Entwicklungsplattform realisiert. Die Windows-Hilfe-Dateien werden mit dem Programm *Help Magician* erstellt. Der *Help Magician* erstellt die Hilfedatei im Rich-Text-Format (RTF) sowie das Help-Projekt-File automatisch und ruft den Help-Compiler unter Windows auf. RTF-Dateien können für jede andere Hardware-Plattform von *OpenUI* über eine Konvertierung in die entsprechenden Hilfesysteme umgewandelt werden. Für Fenstersysteme, die kein Hilfesystem zur Verfügung stellen, wird das "OpenUI Default Help System" (ODHS) angeboten.

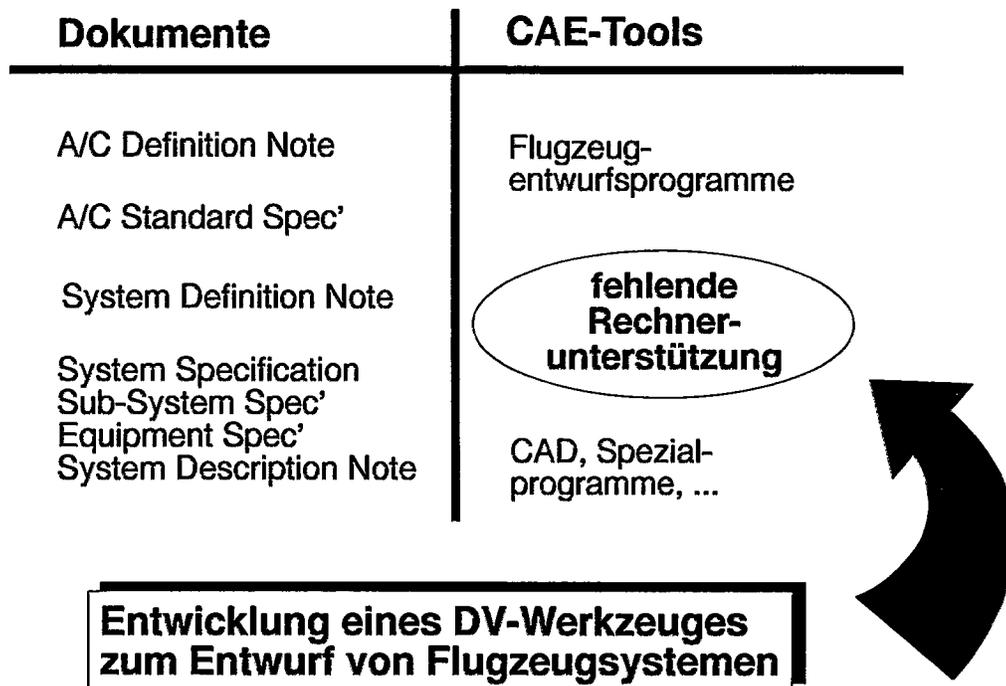
Das Plotprogramm *GNU PLOT* wird von der Oberfläche der CAE-Programm-Module als selbständiges Programm gestartet. Die Plot-Daten und die Plot-Kommandos werden dabei vor dem Start von *GNU PLOT* in entsprechende Dateien geschrieben.

LITERATUR

- [1] Kossira, H.; Heinze, W.; Bardenhagen, A.: *Entwurfswerkzeug PRADO, Entwicklungsstand, Anwendungen und zukünftige Perspektiven*, (DGLR-Workshop: Entwicklungswerkzeuge zur Flugzeugauslegung, Hamburg, 13. Nov. 1995), DGLR, Bonn, 1995.
- [2] Kranz, O.; Haberland, C.: *Visual Capda - Einsatz innovativer Informationstechnologie für den Entwurf und die Analyse von Verkehrs- und Transportflugzeugen*, (DGLR-Workshop: Entwicklungswerkzeuge zur Flugzeugauslegung, Hamburg, 13. Nov. 1995) DGLR, Bonn, 1995.
- [3] DARCorp: *Advanced Aircraft Analysis, Design, Analysis and Research Corporation*, 120 East Ninth Street, Suite 2, Lawrence, Kansas, 66044, USA. - Produktinformation.
- [4] Scholz, D.: *Betriebskostenschätzung von Flugzeugsystemen als Beitrag zur Entwurfsoptimierung*, (DGLR-Jahrestagung, Bonn, 26. - 29. Sept. 1995) DGLR, Bonn, 1995.
- [5] Scholz, D.: *Computer Aided Engineering zur Auslegung von Flugsteuerungs- und Hydrauliksystemen*, (10. Fachtagung Hydraulik und Pneumatik, Dresden, 5./6. Okt. 1995) Technische Universität Dresden, 1995.
- [6] Scholz, D.: *Development of a CAE-Tool for the Design of Flight Control and Hydraulic Systems*, (AEROTECH 95, Birmingham, 17. - 19. Okt. 1995) Institution of Mechanical Engineers, London, 1995.
- [7] Busch, D.A.; Aldana, J.F.: *An Approach to Aircraft Subsystem Design and Integration*, (12th Digital Avionics Conference, 25. - 28. Okt. 1993) Institution of Electrical and Electronics Engineers (IEEE) and American Institute of Aeronautics and Astronautics (AIAA), 1993.
- [8] Hübner, Wolfgang: *Entwurf Graphischer Benutzerschnittstellen*, Springer, 1990.

Zitiervorschlag:

SCHOLZ, Dieter, 1995. Computerunterstützte Auslegung von Flugzeugsystemen am Beispiel der Flugsteuerung. In: SCHOLZ, Dieter, SCHMITT, Dieter (Hrsg.). Entwicklungswerkzeuge zur Flugzeugauslegung. Workshop, DGLR-Fachausschuss S2.1 Starrflügelssysteme (TU Hamburg-Harburg, Arbeitsbereich Flugzeugsystemtechnik, Technologiezentrum Hamburg-Finkenwerder, 13. November 1995). Verfügbar unter: <https://hdl.handle.net/20.500.12738/12639>.



- ➔ Aufbauen auf den Daten der A/C Definition Note
- ➔ DV-Werkzeug gestalten in Anlehnung an Flugzeugentwurfsprogramme

Bild 1: Bedarf für eine Rechnerunterstützung bei der Auslegung von Flugzeugsystemen

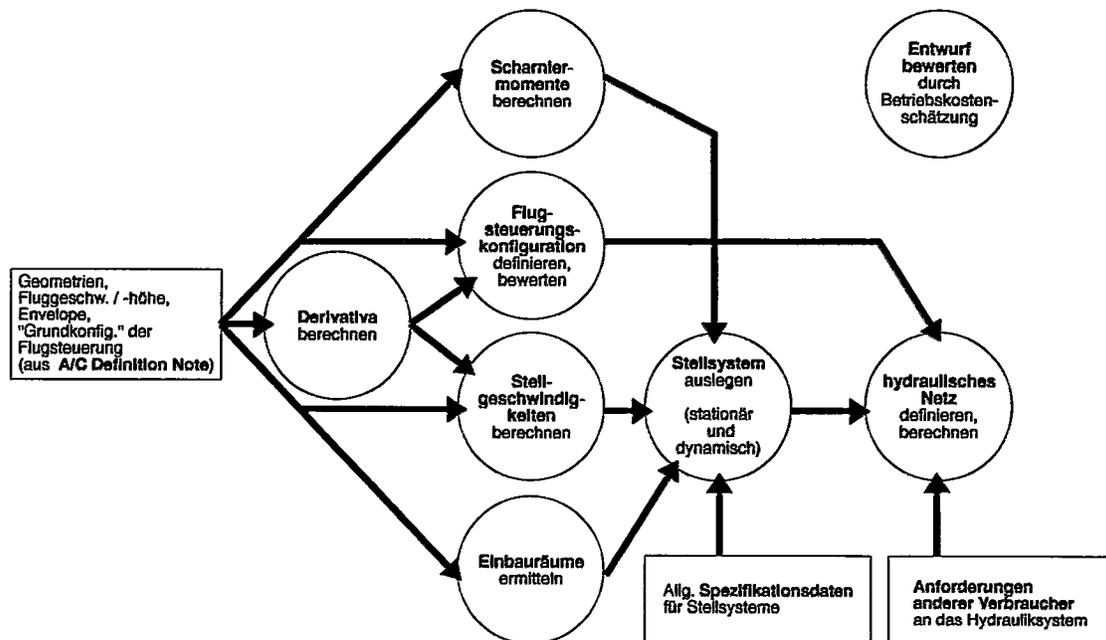


Bild 2: Module des DV-Werkzeuges zur Auslegung von Flugsteuerungs- und Hydrauliksystemen

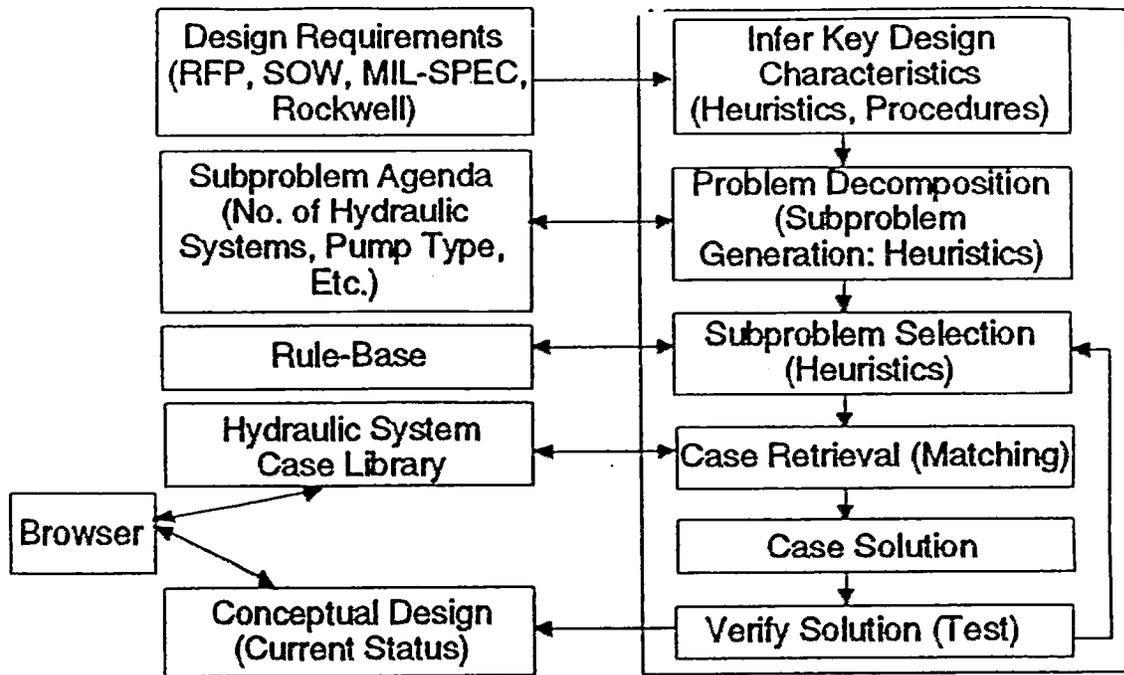


Bild 3: Rockwell International: Computer Aided Engineering for Utility System Development. Use of a "Hybrid Case- and Rule-Based Reasoning Design System"

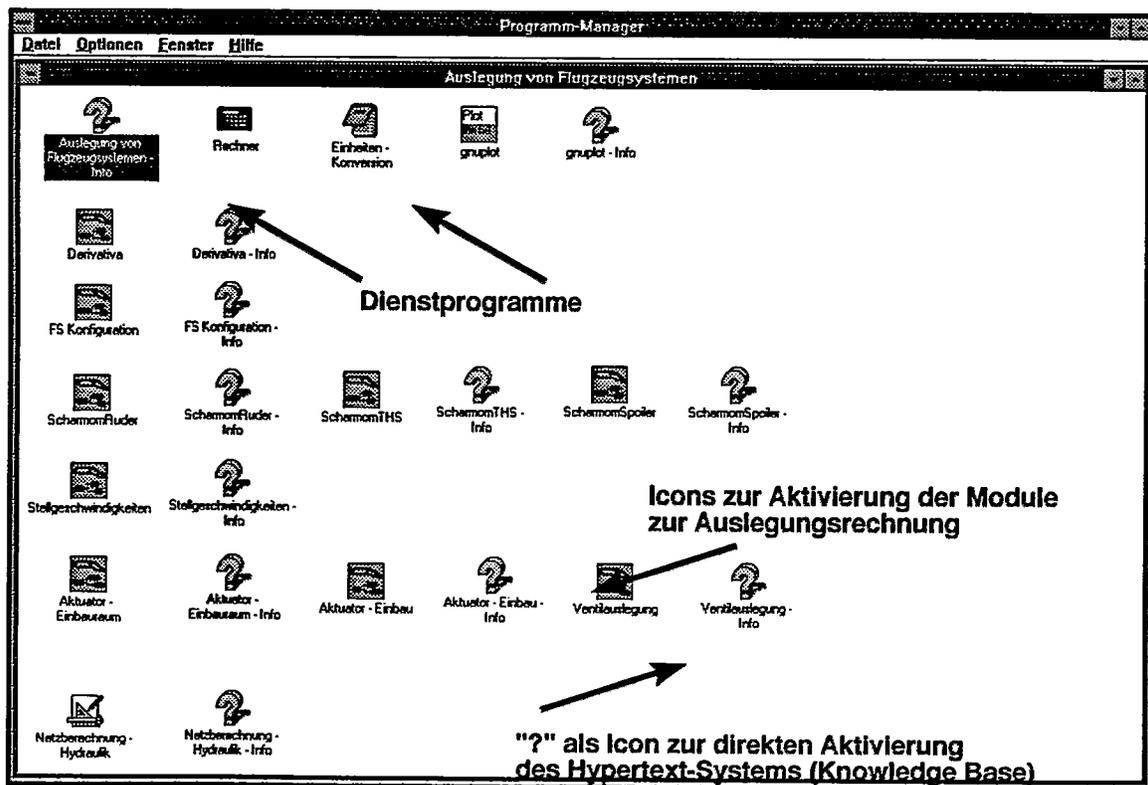
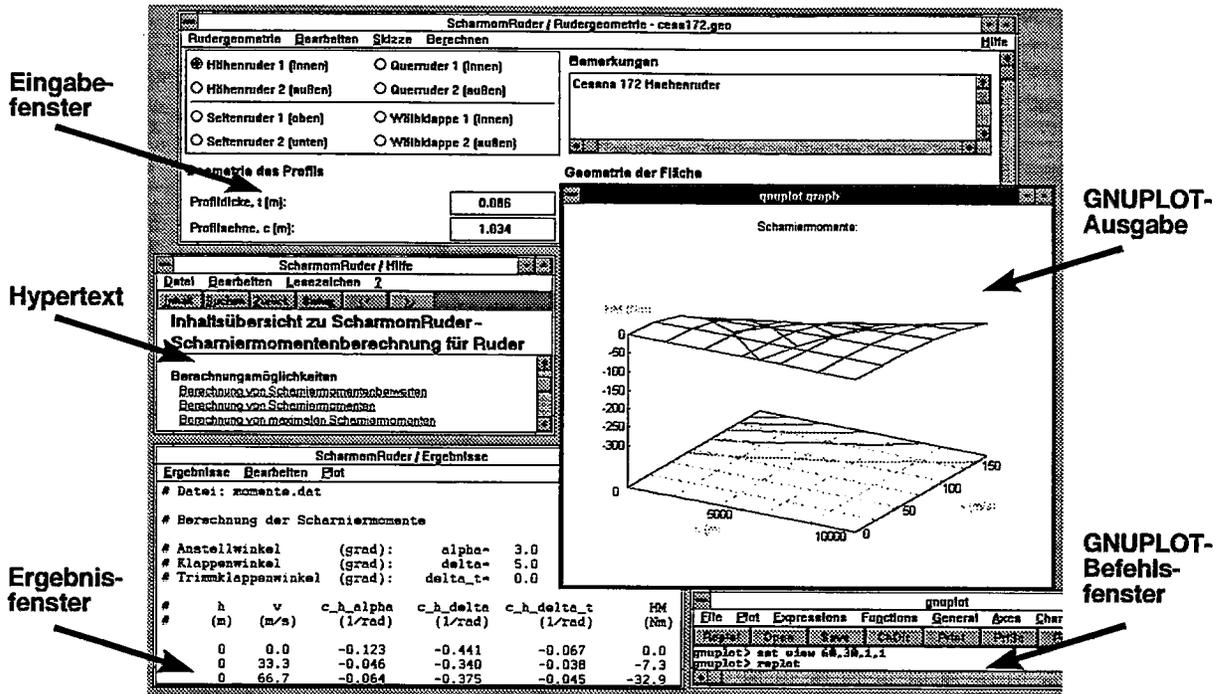


Bild 4: Workplace der Module zur Auslegung von Flugzeugsystemen (hier: Auslegung von Flugsteuerungs- und Hydrauliksystemen)



- Anwender benötigt keine weitere Software
- Qualitätssicherung: - ISO 9241, DIN 66234
- Universelle Datenbasis
- Programmierrichtlinie

Bild 5: "Momentaufnahme" der Oberfläche zum DV-Werkzeug am Beispiel der Scharniermomentenberechnung

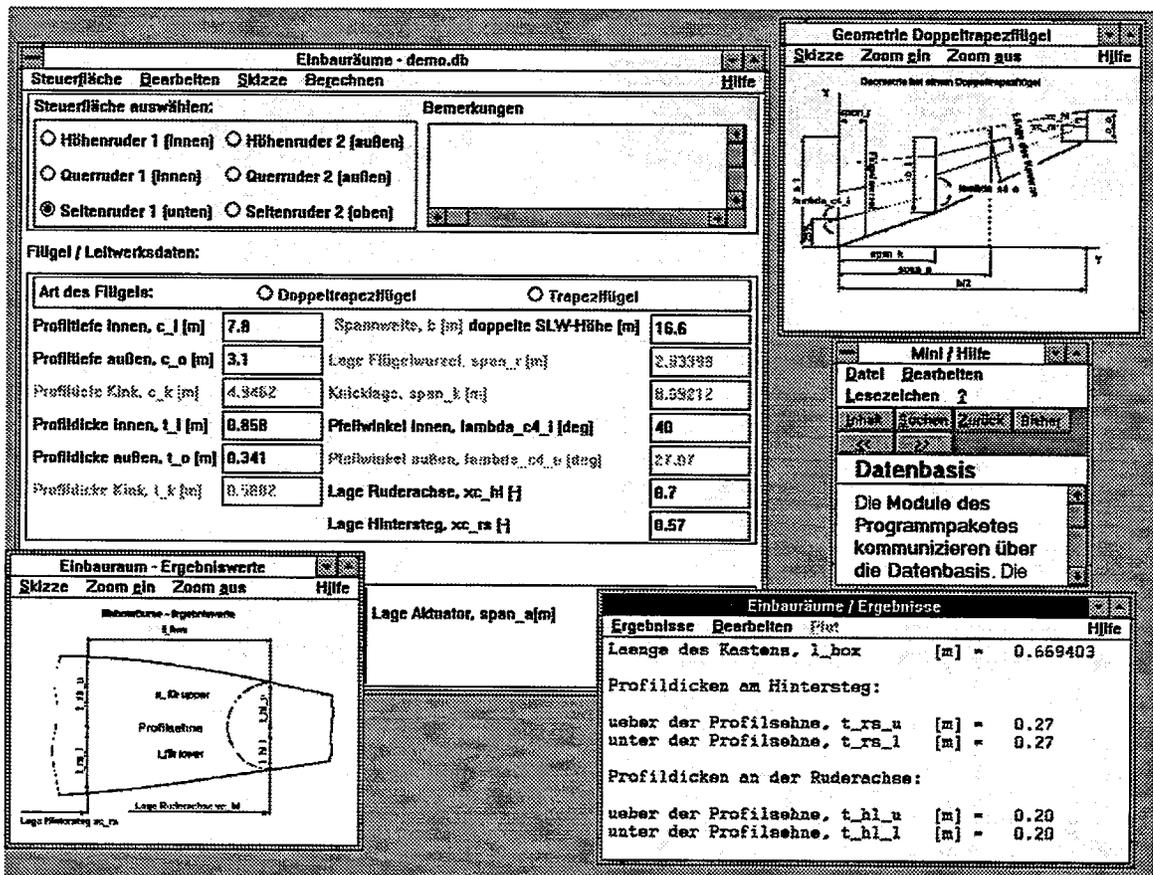


Bild 6: "Momentaufnahme" der Oberfläche zum DV-Werkzeug am Beispiel der Einbauräumeberechnung

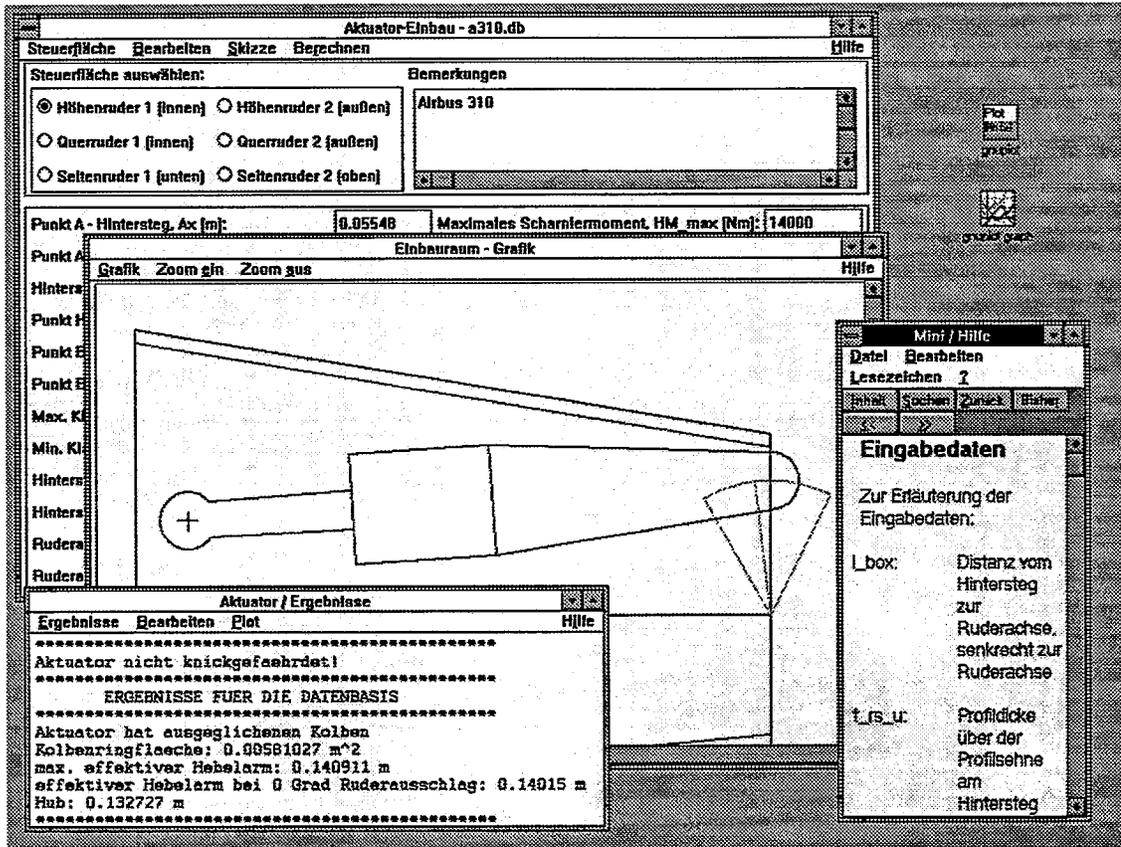


Bild 7: "Momentaufnahme" der Oberfläche zum DV-Werkzeug am Beispiel Aktuator - Einbau

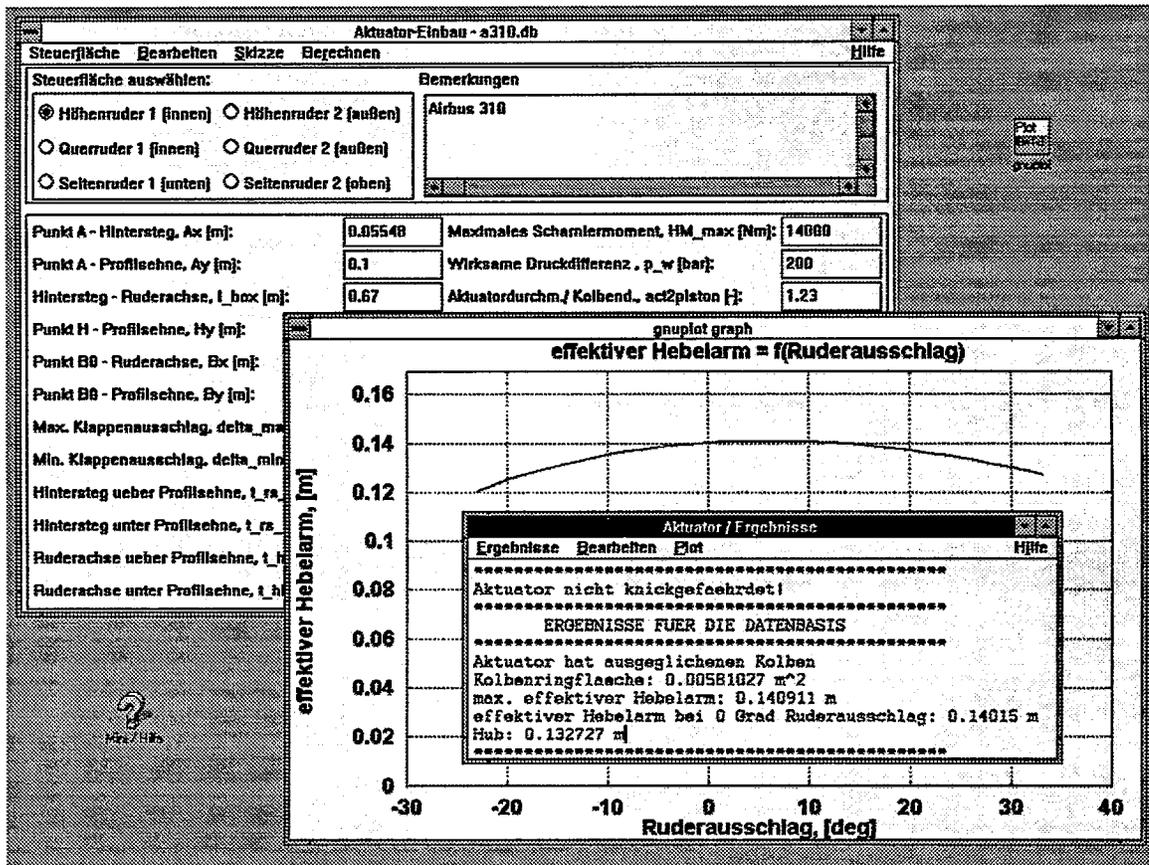


Bild 8: "Momentaufnahme" der Oberfläche zum DV-Werkzeug am Beispiel Aktuator - Einbau

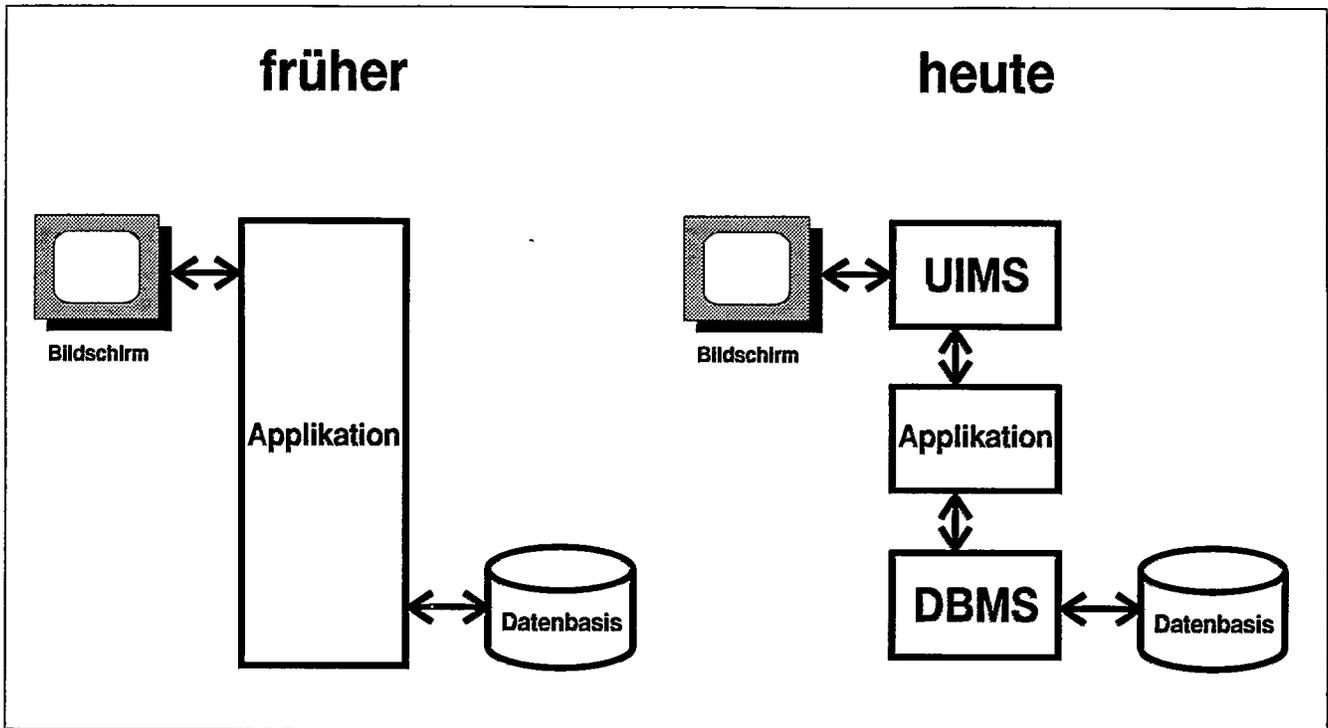


Bild 9: Programmstruktur: früher und heute

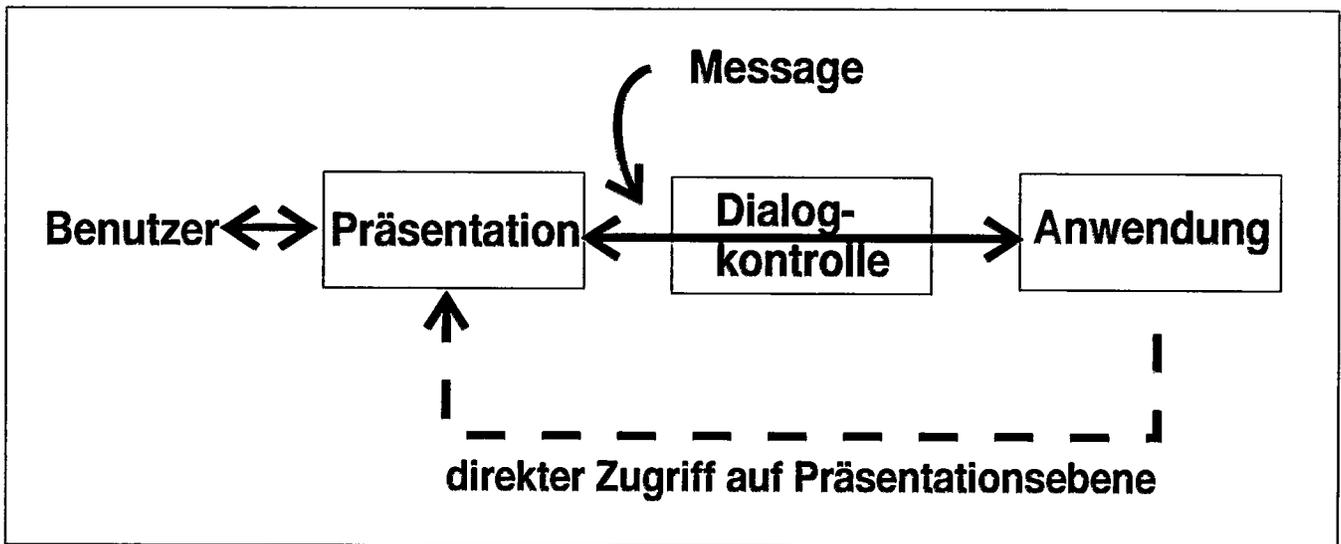


Bild 10: Das Seeheim-Modell: Strukturierung der Aufgabe von User Interface Management Systems

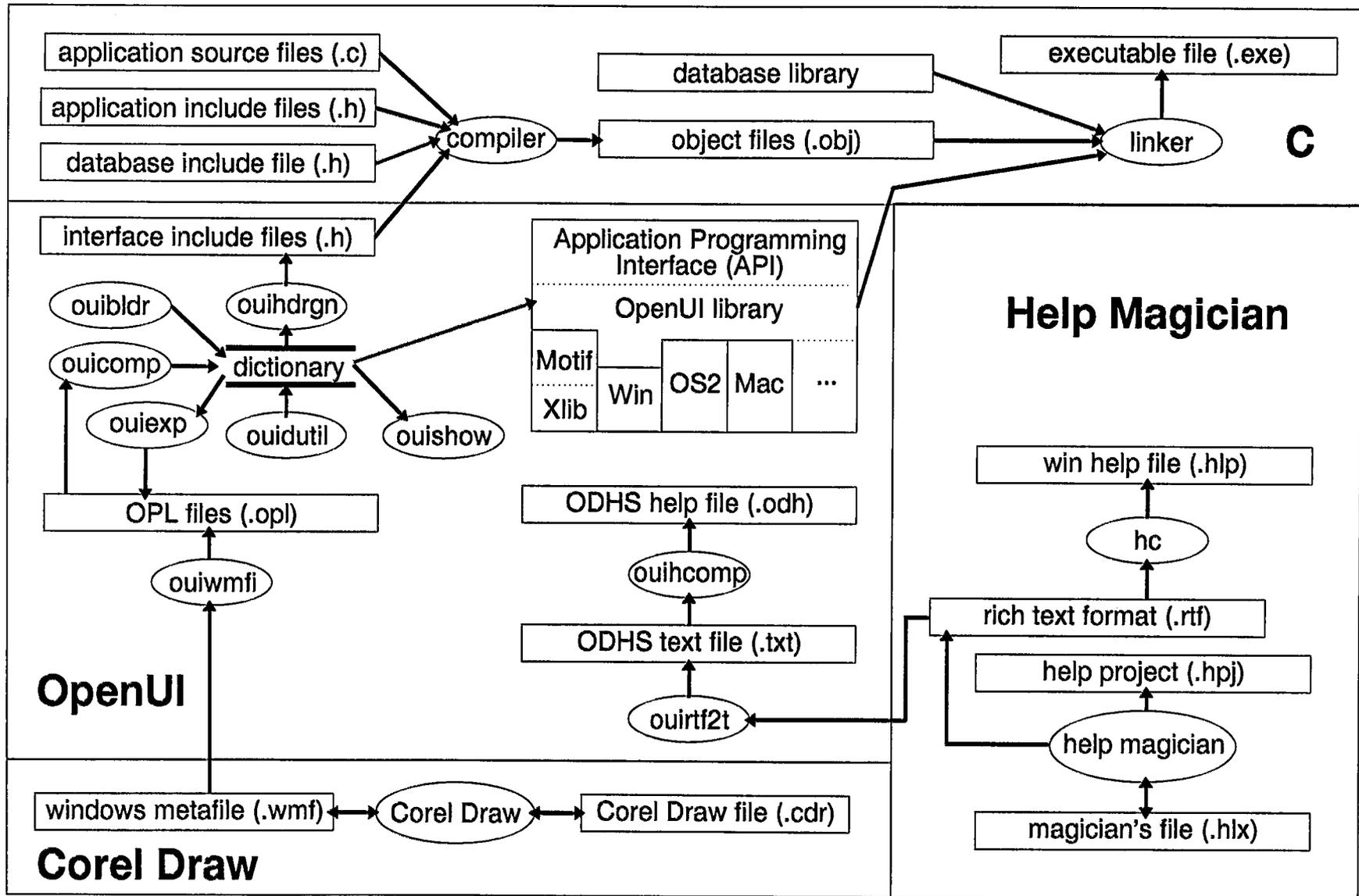


Bild 11: Programmierumgebung