

Schlußbericht

Zuwendungsempfänger:

Deutsches Zentrum für Luft- und Raumfahrt (DLR)

Förderkennzeichen:

TK 0601/3

Vorhabenbezeichnung:

Verbundprojekt: „Verteilte Entwicklung am Beispiel des Flugzeugentwurfs (EFENDA)“

Laufzeit des Vorhabens:

01.03.1996 - 28.02.1998

Berichtszeitraum:

01.03.1996 - 28.02.1998

Inhaltsverzeichnis	Seite
1.0 Aufgabenstellung	2
2.0 Voraussetzungen	5
3.0 Planung und Ablauf des Vorhabens	6
3.1 Schaffung der erforderlichen Infrastruktur seitens des DLR, WP1	6
3.2 Installation von COVISE beim DLR, WP2	6
3.3 Implementierung von MegaCads und FLOWer in COVISE, WP4	7
3.3.1 Implementierung von MegaCads	7
3.3.2 Implementierung von FLOWer	12
3.4 Validierung der Funktionalitäten von MegaCads und FLOWer, WP4	15
3.5 CLIC-3D Implementierung in FLOWer für heterogenen Rechnerverbund, WP5	18
3.6 Schnittstellendefinition für interaktive Prozeßsteuerung in COVISE, WP6	18
4.0 Wissenschaftlicher und technischer Stand vor Beginn des Projektes	20
5.0 Zusammenarbeit mit anderen Stellen	21
6.0 Erzielte Ergebnisse	22
7.0 Voraussichtlicher Nutzen	24
8.0 Fortschritt von anderen Stellen	25
9.0 Veröffentlichungen	25
10.0 Literatur	26

1.0 Aufgabenstellung

Der gegenwärtige Wettbewerbsdruck in der Flugzeugentwicklung verlangt nach einer Reduzierung des zeitlichen und finanziellen Aufwands für aufwendige Parameter- und Konfigurationsstudien. Die numerische Simulation besitzt hierfür ein großes Potential, obgleich sie die teuren Windkanalexperimente in absehbarer Zeit nicht verdrängen wird. Diese werden weiterhin für die Validierung berechneter Ergebnisse und für weitere Verfeinerungen des Designs ihren Stellenwert besitzen.

In immer früheren Stadien des Entwicklungszyklus zukünftiger Flugzeugentwicklungen müssen immer genauere Vorhersagen der Güte der Konfiguration getroffen werden. Der Aerodynamik kommt dabei eine besondere Bedeutung zu, denkt man zum Beispiel an die Vorhersage des Widerstandes, der maßgeblich die Treibstoffkosten beeinflusst. Durch die komplexer werdenden Problemstellungen, wie z.B. die Optimierung von Flugzeugkomponenten unter Berücksichtigung von Wechselwirkungen mit anderen Komponenten des Flugzeugs, instationärer dreidimensionaler Strömungsvorgänge und Wechselwirkungen zwischen Strömung und Struktur steigen schon in naher Zukunft die Anforderungen an die Rechnerprogramme, Rechnerleistung und die Arbeitsmethodik stark an. Die Aerodynamik ist eine Disziplin, die zu den Hauptnutzern numerischer Werkzeuge auf Hochleistungsrechnern im interdisziplinären Prozeß der Flugzeugentwicklung gehört.

Mit Unterstützung vorausgegangener Technologie-, Soft- und Hardwareentwicklungen sollte im Rahmen des BMBF Projektes EFENDA die Effizienz der Bearbeitung von Teilaufgaben der Flugzeugentwicklung deutlich gesteigert werden. Dabei ist gerade die aerodynamische Simulation äußerst rechenzeitintensiv und verlangt daher eine optimale Ausnutzung der Rechnerressourcen durch Metacomputing. Durch die sinnvolle Nutzung der in Deutschland vorhandenen Datenautobahnen (DFN / B-WiN) werden geographisch verteilte Hard- und Software Ressourcen interaktiv nutzbar gemacht, was zu einem erheblichen Zeit- und Kapazitätsvorteil führt. Potentiellen Anwendern, d.h. Ingenieure unterschiedlicher Disziplinen an unterschiedlichen Standorten, wird somit die Möglichkeit gegeben, hochkomplexe Simulationen schnellstmöglich durchzuführen. Die vollständige Simulation einer Flugzeugumströmung beinhaltet dabei das Preprocessing (Netzgenerierung), die Strömungsberechnung und die Visualisierung und Diskussion der Ergebnisse (Postprocessing). Die Strömungsberechnung erfolgt zweckmäßig als Parallel-Processing auf einem heterogenen Rechnerverbund im Hochgeschwindigkeitsnetz mit interaktiver Prozeßkontrolle und -steuerung. Alle Elemente laufen als verteilte Anwendungen und werden durch Audio- und Videokommunikation multimedial unterstützt.

Die Basis der Simulation bilden das Programmsystem MEGAFLOW und das Postprocessing Programm SURVIS. MEGAFLOW beinhaltet sowohl den Netzgenerator MegaCads als Preprocessor als auch das im BMBF-Vorhaben POPINDA parallelisierte Navier-Stokes Verfahren FLOWer. Für die multimediale Integration der o. g. Komponenten stand die Softwareumgebung COVISE zur Verfügung.

In Anlehnung an den Projektantrag vom 30.10.95 [1] werden die Arbeiten der Partner in sechs Arbeitspakete unterteilt - WP1 bis WP6. Das Institut für Entwurfsaerodynamik des DLR war dabei an folgenden fünf Arbeitspaketen beteiligt:

WP1 - Koordination und Bereitstellung der Netzwerk-Infrastruktur

Die Verbindung von Audio-, Video- und Dateninformation innerhalb einer Kommunikationsverbindung zwischen mehreren Partnern erfordert eine Anpassung der Netzwerkinfrastruktur.

WP2 - Implementierung der COVISE Benutzerumgebung

Die multimediale Benutzeroberfläche COVISE des RUS muß beim DLR auf der neuen Workstation (s. WP1) installiert, konfiguriert und getestet werden.

WP4 - Implementierung von MegaCads und FLOWer in COVISE

Die Verfügbarkeit von MegaCads und FLOWer in COVISE erfordert eine Modifikation der Source-Codes, um eine Kommunikation zwischen COVISE und den Modulen zu ermöglichen. In Abb. 1 ist ein prinzipielles Ablaufdiagramm dargestellt, das die notwendigen Datentransfers erkennen läßt. Die mit MegaCads generierten Netzdaten werden vom Strömungslöser FLOWer benötigt. Dieser berechnet die Strömung in Abhängigkeit zum gewählten Parametersatz. Das Ergebnis wird dem Postprocessing Modul SURVIS zur Visualisierung verfügbar gemacht. Eine ausgedehnte Validierung bildet den Abschluß dieses Arbeitspaketes.

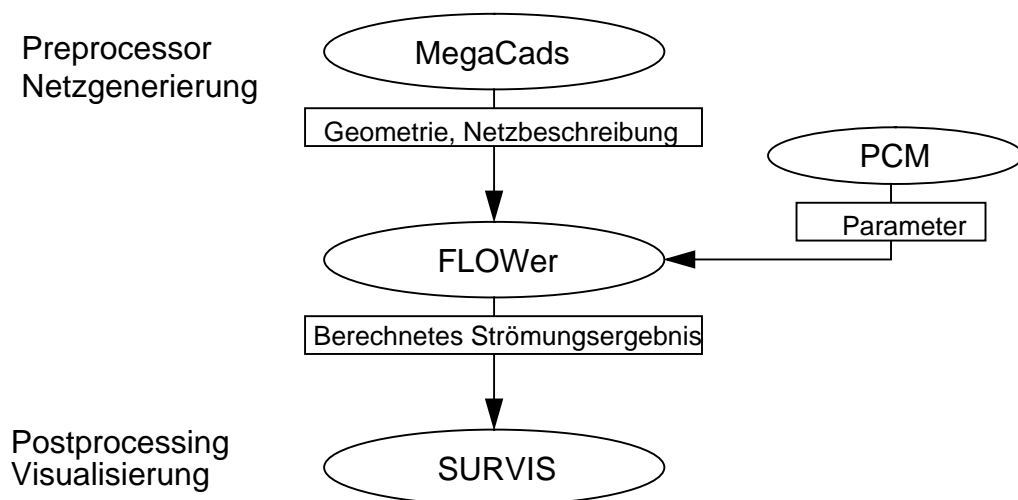


Abb. 1: Prozeßkette der COVISE-Module für EFENDA

WP5 - Anpassung der Parallelisierung von FLOWer für heterogenen Rechnerverbund

Modifikation der CLIC-3D und entsprechende Anpassungen im FLOWer Code, um die parallele Version im heterogenen Rechnerverbund lauffähig zu machen.

WP6 - Anpassung/Verbesserung der interaktiven Prozeßsteuerung in COVISE

In Anlehnung an WP4 sollen geeignete Schnittstellen für die interaktive Benutzung von MegaCads und FLOWer definiert und umgesetzt werden. Speziell für FLOWer soll ein Prozeß-Steuerungs-Modul (PCM) geschaffen werden, das die interaktive Modifizierbarkeit des Parameterdatensatzes zuläßt.

Die nachfolgende Tabelle, Tab. 1, zeigt den Zeitplan für die Aufgaben des DLR. Die Abwicklung der Aufgabenpakete ist nicht direkt an die Hauptmeilensteine des Projekts gekoppelt. Der vollständige Zeitplan sowie die exakte Beschreibung der Meilensteine ist dem Zwischenbericht [2] zu entnehmen.

		1996					1997					1998	
WP	Mon.	04	06	08	10	12	02	04	06	08	10	12	02
WP1													
WP2													
WP3													
WP4													
WP5													
WP6													

Tab. 1: Zeitplan der DLR-Arbeitspakete im EFENDA-Projekt

2.0 Voraussetzungen

Die Dauer des Projektes wurde für 2 Jahre vorgesehen. Projektstart war der 1.3.1996. Die Mittel für einen Wissenschaftlichen Mitarbeiter und die Aufrüstung der erforderlichen Hardware waren bewilligt (s. hierzu WP1, Kapitel 3). Die einzelnen Programmpakete, COVISE (RUS), MegaCads (DLR) und FLOWer (DLR) wurden je nach Entwicklungsstand an die Partner verteilt und installiert. Bei MegaCads handelte es sich dabei um die Version-2.1, bei FLOWer zunächst um die Version 113. Während des Projektverlaufs wurden die Entwicklungsarbeiten vollständig auf das neuere FLOWer-Release 114 umgestellt. Hintergrund dieser Umstellung war die erweiterte Funktionalität in FLOWer 114, insbesondere die verbesserte Turbulenzmodellierung, die eine genauere Wiedergabe physikalischer Phänomene zulässt. Beide Programmpakete wurden mit ausführlichen Benutzerhandbüchern ausgeliefert, die den Benutzer bei der Installation und der Anwendung stark unterstützen [5], [10], [15].

Das im Projektverlauf entwickelte PCM (TUB) zur kooperativen, interaktiven Steuerung der FLOWer-Eingabeparameter, wurde den Partnern erstmals im April '97 als Prototyp vorgestellt. Die folgenden Releases wurden den anderen Paketen entsprechend den Partnern zur Verfügung gestellt.

3.0 Planung und Ablauf des Vorhabens

Nach anfänglicher Verzögerung der Projektarbeiten durch den Bewilligungsvorgang und damit verbundenen administrativen Zusatzarbeiten und Engpässen bei der Personaleinstellung im universitären Bereich der Projektpartner konnte der ursprünglich geplante Projektstatus zum Zwischenbericht (Feb. 1997) nahezu wieder aufgeholt werden [2]. Dazu gab es leichte Anpassungen der Meilensteine im Zeitplan.

3.1 Schaffung der erforderlichen Infrastruktur seitens des DLR, WP1

Die Verbindung von Audio-, Video- und Dateninformation innerhalb einer Kommunikationsverbindung zwischen mehreren Partnern erforderte eine Anpassung der Infrastruktur im DLR. In einem ersten Schritt wurde ein spezieller EFENDA-Arbeitsplatz angeschafft und eingerichtet. Das System ist charakterisiert durch:

- SGI-Workstation
- MIPS R10000 Prozessor, 195 MHz
- Hauptspeicher: 768 Mbytes
- Iris Audio Prozessor
- Solid Impact Grafikkarte
- Indigo2 Video, Indycam

Die Leistungsfähigkeit dieses Arbeitsplatzes konnte in mehreren COVISE-Sitzungen erfolgreich getestet werden.

Der bei der Kommunikation mit anderen Partnern entstehende Engpaß der Datenübertragung durch das Netzwerk implizierte den zweiten Schritt der erforderlichen Maßnahme. Die Schaffung eines 8 Mbit/s Breitbandanschlusses an das B-WiN des DFN erfolgte im April 1997, wobei es sich hier zunächst um einen zentralen Anschluß an den DLR Hauptsitz in Köln-Porz handelte. Die Breitbandverbindung zwischen Köln und Braunschweig wurde zum Ende des Jahres 1997 bzw. Anfang 1998 geschaffen.

3.2 Installation von COVISE beim DLR, WP2

Durch ein vom RUS erarbeitetes Installationsskript konnte die Erstinstallation von COVISE problemlos innerhalb weniger Minuten erfolgen. Alle weiteren Versionen, derzeit liegt die Version 4.3 vor, waren Updates, die ohne Installationsskript eingespielt werden konnten. Die Installation wurde stets auf der im WP1-Arbeitspaket beschriebenen SGI-Plattform durchgeführt.

Der Leistungsumfang von COVISE konnte durch die frühzeitige Auslieferung des Benutzerhandbuches bereits vor Auslieferung der ersten Version eingehend studiert werden. Allerdings zeigten sich hier Unzulänglichkeiten in der Aktualität des Handbuches und der für den Zeitpunkt benutzten COVISE-Version [9]. Die Integration neuer Software in die kooperative Ober-

fläche war somit nur unter erschwerten Bedingungen möglich

Die anschließenden Tests zur Überprüfung der Funktionalitäten von COVISE auf der DLR-Plattform führten zu einem insgesamt positiven Eindruck. Bestehende Beschränkungen sowie Stabilitätsprobleme des Systems wurden an das RUS weitergeleitet.

3.3 Implementierung von MegaCads und FLOWer in COVISE, WP4

Aufbauend auf dem im DLR entwickelten Netzgenerator MegaCads und dem Strömungslöser FLOWer wurde im Rahmen des BMBF-Luftfahrtforschungsprogramms das Verbundvorhaben MEGAFLOW initiiert. Ziel des Vorhabens ist die Erstellung eines qualitätsgesicherten und effizienten Programmsystems zur Strömungssimulation um komplette Verkehrsflugzeuge in Reiseflug- und Start- und Landekonfiguration [4].

3.3.1 Implementierung von MegaCads

Das Netzgenerierungssystem MegaCads

MegaCads ist ein strukturierter Mehrblock-Netzgenerator mit eingeschränkter CAD-Funktionalität. MegaCads steht für: **M**ultiblock **E**lliptic **G**rid **G**eneration **a**nd **C**AD **S**ystem. Aufgrund des hohen manuellen Aufwands bei der Netzgenerierung verfolgt MegaCads ein parametrisches Prinzip. Dies ermöglicht die wiederholte Anwendung von bereits durchgeführten Netzgenerierungsschritten für geänderte Geometrien [6],[7].

Die in MegaCads verfügbaren Basis-CAD-Funktionalitäten erlauben dem Benutzer Netzgenerierungsschritte mit hohem Freiheitsgrad durchzuführen. Die Bandbreite der implementierten Funktionalitäten reicht von der Konstruktion einzelner Kurven im Raum, über Flächen bis hin zu beliebig gearteten Blöcken. Ebenso existieren Techniken zum Teilen, Verbinden und Kopieren der genannten Elemente. Neben den CAD-Funktionen existieren spezielle Netzgenerierungsalgorithmen für algebraische und elliptische Netze.

Bei der in COVISE integrierten MegaCads-Version handelt es sich um das inzwischen vorletzte Release, MegaCads-2.1. Abb.3 gibt einen Überblick über die Bedienoberfläche des Netzgenerators.

Der mit 1 gekennzeichnete Bereich ist die Menüleiste, über die sämtliche Gruppenfunktionalitäten angesprochen werden können. Diese Gruppen werden detailliert als widgets im Gebiet 2 dargestellt. Durch Anwählen einer weiteren Gruppe müssen die gezeigten widgets neu organisiert werden, d.h. die der Gruppe zugehörigen widgets eingeblendet werden. Das eigentliche Arbeitsfenster, Bereich 3, stellt die sogenannten Entities der einzelnen Netzgenerierungsschritte dar. Entities sind im allgemeinen Linien, Flächen oder Blöcke. Alle Entities in diesem Fenster können translatiert, vergrößert oder verkleinert und rotiert werden. Desweiteren ist es möglich, einzelne Entities über die Entity-Liste, Bereich 4, ein- oder auszublenden. Das Anwählen einer oder mehrerer Entities wird durch eine besondere Farbgebung in der Liste und im Arbeitsfenster sichtbar gemacht. Das Ein- und Ausblenden sowie das Selektieren wird über die Funktionen im Bereich 5 unterstützt. Die Visualisierung wird durch zahlreiche Steuerelemente im Bereich 6 vereinfacht. Hier finden sich Möglichkeiten, alte Ansichten zurückzuholen, Standardansichten anzuwählen, alle Entities im Arbeitsfenster darzustellen und natürlich die bekannten Grundelemente der Ansicht, Rotation, Translation und Vergrößerung/Verkleinerung

zu benutzen. Ein Kommentarfenster, Bereich 7, informiert den Nutzer über den Ablauf ange-
wählter Funktionen.

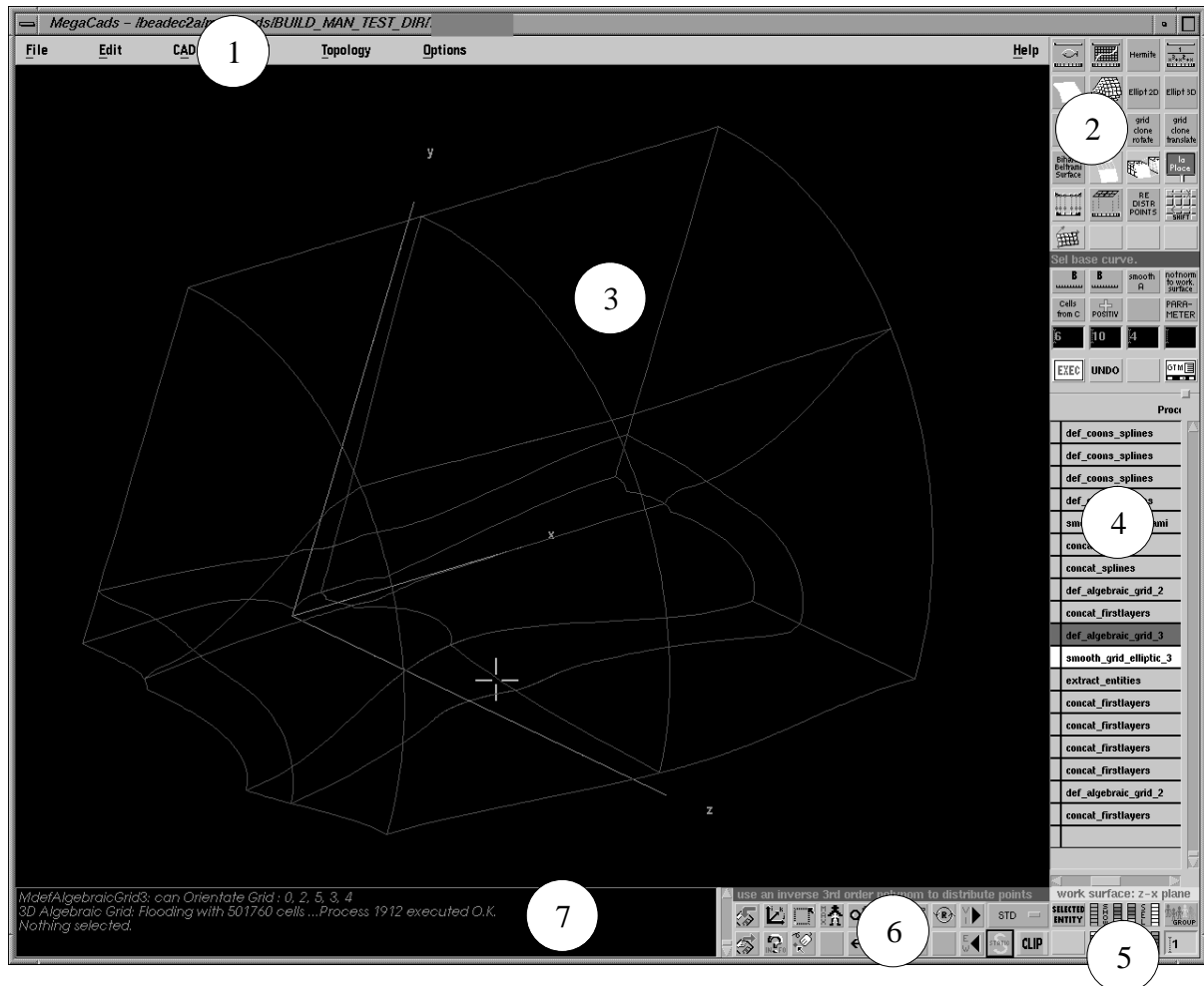


Abb. 3: MegaCads Bedienoberfläche (GUI - Graphical User Interface)

Die hier in aller Kürze vorgestellten MegaCads-Funktionalitäten müssen für ein kooperatives Arbeiten vollständig zur Verfügung stehen. Ihre Vielfalt und die Notwendigkeit, die Details der MegaCads-Bedienung bei allen Partnern einer MegaCads-Sitzung sichtbar zu machen, lassen den Aufwand der Integration erkennen. Insgesamt mußten über 100 Funktionalitäten für das kooperative Arbeiten aufbereitet werden. Es ist absolut notwendig, daß alle Informationen zugänglich sind, die die Arbeitsweise des aktiven Netzgenerators bei allen Partnern erkennen läßt. Die Partner könnten keinen realen Eindruck von den Netzgenerierungsschritten gewinnen, wenn z. B. die Selektion und die damit verbundene Farbänderung einer Entity nicht bei allen Teilnehmern visualisiert werden würde.

Vorgehensweise bei der MegaCads Integration

Die Integration sieht vor, den Netzgenerator bei allen Partnern über COVISE starten zu können

und anschließend jeden notwendigen Schritt zur Fertigstellung des Rechnetzes an die Partner weiterzugeben. Dies bedeutet, daß sowohl Konstruktionsschritte als auch Schritte zur Visualisierung weitergegeben werden müssen (s. o.).

Die generelle Vorgehensweise bei der Integration des MegaCads-Moduls baut auf das Versenden von Informationen auf. Führt der Initiator (Master) einer Sitzung eine Aktion aus, wird diese Aktion verpackt und an alle beteiligten Partner (Slaves) verschickt. COVISE bietet hier zwei Möglichkeiten. Es können COVISE Datenobjekte definiert werden, die den inhärenten MegaCads Datenstrukturen entsprechen und somit bequem zu verschicken sind. Alternativ können diese Strukturen in Character-Arrays umgewandelt und über sogenannte RENDER-Messages verschickt werden. Die Entscheidung fiel, nicht zuletzt aufgrund der Quantität der zu definierenden Objekte (pro MegaCads Funktionalität ein Datenobjekt), auf die zweite Variante. Die generelle Vorgehensweise wird in Abb. 4 an einem Beispiel zur Generierung einer Kurve verdeutlicht.

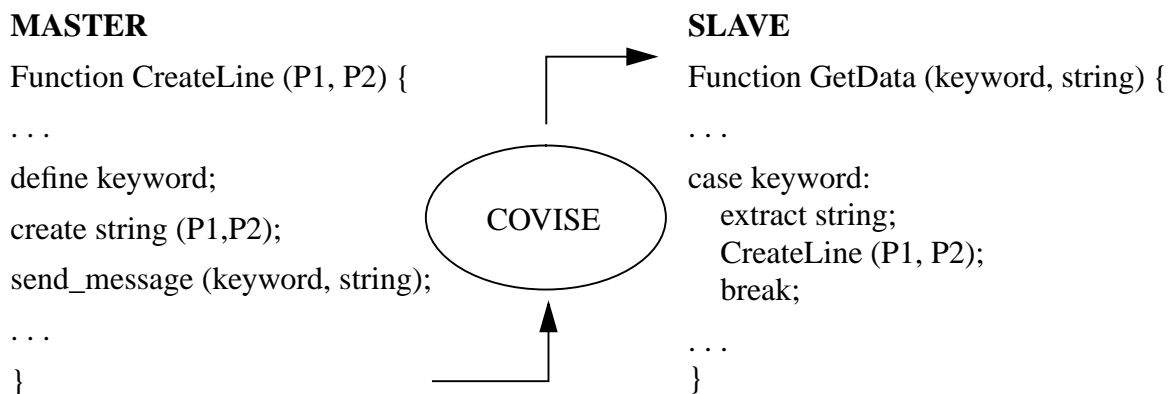


Abb. 4: Vorgehensweise des Datenversendens für MegaCads (Pseudo-Code)

Der Master generiert eine Linie und spricht dabei die Funktion CreateLine in MegaCads an. Die Argumente der Funktion bestehen aus einem Start- und Endpunkt (P1, P2). Nach erfolgreicher Ausführung der Funktion wird ein Keyword vergeben und die für die Funktion essentiellen Parameter in ein Character-Feld geschrieben. Nach Versenden stehen beide Informationen dem Slave zur Verfügung, der die Daten extrahiert und den Prozeß startet. Das Verschicken der Daten erfolgt nur in eine Richtung - vom Master zum Slave. Eine Kontrolle über die korrekte Durchführung des Funktionsaufrufes erfolgt zweckmäßig über die audio-visuelle Verbindung der Teilnehmer der kooperativen Sitzung und über die Meldungen im Kommentarfenster der MegaCads-Oberfläche.

Die Erweiterung des MegaCads-Codes zur kooperativen Nutzung in COVISE wurde in folgende Entwicklungsstufen aufgeteilt:

- Implementierung der Funktionalitäten zur kooperativen Visualisierung ohne Veränderung der Ursprungsdaten. In dieser Entwicklungsstufe soll die gemeinsame Visualisierung eines bereits fertiggestellten Netzes zwischen mehreren Partnern einer

kooperativen Sitzung möglich sein. Eine Erweiterung oder Veränderung des Rechnernetzes ist nicht gefordert.

- Verfügbarmachung sämtlicher in MegaCads vorkommender Funktionalitäten zur Veränderung oder Modifikation des Rechnernetzes. Diese Stufe ermöglicht bereits, das vollständige Rechnernetz kooperativ zu generieren. Die dafür in MegaCads vorhandenen Funktionalitäten sind vollständig kooperativ einsetzbar.
- Kooperative Gestaltung des Graphical User Interfaces (GUI), inklusive des Mauszeigers. Erst die Anpassung der GUIs vermittelt den Eindruck einer kooperativen Sitzung: Menüpunkte werden gemeinsam durchgeblättert, das Selektieren von Entities im Arbeitsfenster oder in der Entityliste wird durch einheitliche Farbgebungen gekennzeichnet, einzelne Buttons werden an den Prozeß angepaßt und der Mauszeiger wird synchronisiert.
- Erstellung eines Konzeptes zur Erfüllung der Sicherheitsanforderungen der Industrie. Diese letzte notwendige Entwicklungsstufe wurde im Projektverlauf definiert und ausschließlich auf konzeptioneller Ebene bearbeitet. Eine praktische Umsetzung fand aus zeitlichen Gründen nicht statt.

Die ersten drei Punkte werden nach dem im vorigen Abschnitt besprochenen Prinzip der Keyword-Vergabe und dem Verpacken/Entpacken der funktionsabhängigen Parameter umgesetzt. Zu diesem Zweck wurden Module geschaffen, die prozeßabhängige Keywords vergeben und die spezifischen Prozeßparameter verpacken. In weiteren Modulen werden diese Keywords identifiziert und die Parameter keywordspezifisch entpackt.

Für das Versenden und Empfangen der Daten in der COVISE-Umgebung werden entsprechende Funktionen vorgehalten. Der Aufruf

```
Covise::send_render_message(cKey,cMsg)
```

verschickt die Daten, während die Aufrufe

```
Covise::get_render_keyword() und Covise::get_render_data()
```

die Daten wieder empfangen. Auf der Slave-Seite müssen die Prozeßdaten nach Empfang entschlüsselt werden. Hierzu dient das keyword cKey, das bereits Aufschluß über den Prozeß gibt. Somit können die Prozeßdaten keyword-spezifisch aus dem Character-Feld extrahiert werden. Nachdem die Prozeßdaten auf der Slave-Seite zusammengestellt wurden und keine Fehler aufgetreten sind, wird der Prozeß in MegaCads beim Slave gestartet.

Der Aufwand der hier beschriebenen Vorgehensweise für die Visualisierung und die GUI-Anpassung auf der einen Seite und der MegaCads CAD- und Netzgenerierungsfunktionalitäten auf der anderen Seite differierte bei der Integration. Zum einen haben die GUI- und Visualisierungsprozesse kein einheitliches, übergeordnetes Prozeßdatenformat. Zum anderen existiert kein gemeinsamer Knotenpunkt wie bei den Funktionalitätensteuerung, so daß mehrere MegaCads-Funktionen für die COVISE-Integration angepaßt werden mußten.

COVISE bietet verschiedene Modultypen für die Integration neuer Softwareprodukte an. Der gewählte Modultyp „Renderer“ für die MegaCads-Integration in COVISE besitzt den Nachteil,

daß keine Daten von MegaCads an das Folgemodul FLOWer weitergegeben werden können (vgl. Kap. 6: Erzielte Ergebnisse). Da zukünftige MegaCads Versionen in der Lage sein werden, sämtliche für die FLOWer Rechnung benötigten Daten bereit zu stellen, wurde bereits mit der Entwicklung eines weiteren MegaCads-Moduls vom Typ „Generic“ begonnen. Die Arbeiten zur Umstellung des MegaCads-Moduls auf diesen neuen COVISE Modultyp wurden auf konzeptioneller Ebene abgeschlossen, aufgrund des zeitlich begrenzten Rahmens jedoch noch nicht vollständig umgesetzt. Hierzu sei angemerkt, da es sich nicht um ein im EFENDA Projekt definiertes Arbeitspaket, sondern um eine vom EFENDA-Team selbstdefinierte Zusatzarbeit handelt. Das Konzept wird an dieser Stelle nur in seiner groben Struktur vorgestellt. Genauere Inhalte sind [11] zu entnehmen. Das Konzept sieht folgenden Ablauf vor: 1) Definition des Ausgabedatenports mit der add_port Funktionalität von COVISE 2) Festlegen des Ausgabeformates 3) Anlegen des Datenobjektes beim Master 4) Erweiterung der Export-Funktionalität in MegaCads 5) Buffer für Datenobjekt anlegen 6) Anstoßen der Modulpipeline über Message 7) Schreiben des Datenobjektes 8) Finished-Message nach Beendigung des Schreibvorgangs.

Ein letzter Punkt betrifft die Notwendigkeit der Geheimhaltung der Daten bei gemeinsamer Nutzung des Netzgenerators MegaCads. Diese Aufgabe wurde im Verlauf des Projektes aus aktuellem industriellen Anlaß spezifiziert, mit dem Ziel, Konzeptvorschläge für ein eventuelles Folgeprojekt zu erarbeiten. Meilensteine wurden nicht definiert. Das Konzept beinhaltet drei Lösungsvorschläge, die im wesentlichen mit RUS diskutiert wurden [11]. Eine Entscheidung für einen der Vorschläge wurde nicht getroffen.

Konzeptvorschlag 1: Derzeit müssen die für MegaCads erforderlichen Inputdaten lokal vorliegen, d.h. jeder der Partner muß die Informationen der zu vernetzenden Geometrie vorher via ftp kopieren. Im wesentlichen liegt das am Fehlen eines gemeinsamen Speicherbereichs, auf den die Partner unter COVISE mit unterschiedlich definierten Zugriffsrechten verfügen können. Da die Geheimhaltung die zu vernetzende Geometrie betrifft und nicht etwa das generierte Netz, besteht hier die Möglichkeit, aus einem gemeinsamen Datenspeicher zu lesen und das Ausschreiben der Daten aus MegaCads bei allen Partnern zu sperren. Lediglich der Initiator einer Sitzung bzw. der Master hat das Recht, Geometriedaten aus MegaCads zu exportieren. Die dafür in MegaCads notwendigen Modifikation würden sich auf einen einzigen Prozeß, der im Moment mit expliziten Pfadangaben arbeitet, beschränken und wäre damit sehr schnell zu realisieren. Eine Beurteilung dieses Ansatzes kann nur durch das RUS erfolgen, das die Expertise der COVISE-Benutzerumgebung hält.

Konzeptvorschlag 2: Eine zweite Möglichkeit konzentriert sich auf die vollständige Überarbeitung der MegaCads-Funktion für den Datenimport. Der Master hat die Daten lokal vorliegen und liest sie mit der bekannten (unveränderten) Funktion für den Datenimport ein. Bei den Partnern (Slaves) ist diese Funktionalität nun soweit verändert worden, daß die Daten nicht von einem lokalen Speichermedium gelesen werden, sondern als vom Master verschickte Messages empfangen werden. An dieser Stelle ist höchste Effizienz an das Verschicken der Messages gefordert, da im Normalfall mehrere Millionen float-Werte zu verschicken sind. Aus jetziger Sicht bietet sich hier die Definition eines geeigneten COVISE-Datenobjektes an, das ohne die oben beschriebene Verschlüsselung zu benutzen ist. Wie im Konzeptvorschlag 1 beschrieben, wird das Ausschreiben der Geometriedaten nur dem Master ermöglicht. Der Aufwand für die notwendige Änderung bzw. die Neugestaltung der Datenimport-Funktion wird als relativ gering eingestuft.

Konzeptvorschlag 3: Eine Alternative zu den Konzeptvorschlägen 1 und 2 wäre die

ausschließliche Übertragung von X- bzw. OpenGL-Befehlen vom Master. Hier sollten keine Sicherheitsbedenken vorliegen, da sich aus diesen Informationen maximal der Bildschirminhalt ableiten läßt, jedoch nicht das eigentliche dreidimensionale Modell. Für diesen Ansatz ist bereits sicher, das die Performance deutlich verschlechtert wird. Zusätzlich würden erhebliche Änderungen in MegaCads notwendig werden, die den Code in seiner gesamten Struktur verändern. Der Zeitaufwand wäre erheblich, weshalb dieser Ansatz aus DLR Sicht nicht favorisiert wird.

3.3.2 Implementierung von FLOWer

Der Strömungslöser FLOWer

Der für den industriellen Einsatz zur reibungsbehafteten Strömungsberechnung entwickelte Flower-Code ist Teil des BMBF-Verbundvorhabens MEGAFLOW. FLOWer [5], [15] löst die Reynolds-gemittelten Navier-Stokes-Gleichungen. Er basiert auf einer Finite-Volumen-Diskretisierung und nutzt den blockstrukturierten Ansatz für die Berechnung komplexer Konfigurationen. Der Basisalgorithmus besteht aus einer zentralen Raumdiskretisierung in Verbindung mit einem expliziten Zeitschrittverfahren. Zur Berechnung stationärer Strömungen kommen verschiedene Beschleunigungstechniken zum Einsatz, von denen das Mehrgitterverfahren aufgrund seiner hohen Effizienz besonders hervorzuheben ist.

Die Parallelisierung des Codes eröffnet zum einen die Möglichkeit, die Antwortzeiten und Rechenkosten für große, praxisrelevante Anwendungen drastisch zu reduzieren. Zum anderen können Aufgabenstellungen der Konfigurationsaerodynamik behandelt werden, die aufgrund ihrer Speicherplatzanforderungen auf Einzelprozessoren nicht durchführbar sind [15]. Im Rahmen des vom BMBF geförderten Verbundprogrammes POPINDA wurde in Zusammenarbeit mit der deutschen Luftfahrtindustrie, der Gesellschaft für Mathematik und Datenverarbeitung (GMD) sowie den Firmen IBM und ORCOM der FLOWer Code für die Anwendung auf parallelen Rechnersystemen weiterentwickelt.

Die Parallelisierung von FLOWer beruht auf dem "Message Passing"-Host-Node-Programmiermodell und einer Grid Partitionierung. Während des POPINDA-Projektes wurde die Parallelisierung durch die Einbindung der "High-Level"-Kommunikationsbibliothek CLIC-3D realisiert. Abhängig von der Rechnerarchitektur kann FLOWer wahlweise mit oder ohne die CLIC-3D-Bibliothek benutzt werden. Der sequentielle FLOWer arbeitet nicht mit dieser Bibliothek, während die parallele Variante mit der Bibliothek gelinkt wird. Die Portabilität des sequentiellen FLOWers erstreckt sich über alle Einzelprozessorrechner bis zu "Shared-Memory"-Parallelrechnern. Die Portabilität des parallelen FLOWers ist durch die Verfügbarkeit einer CLIC-3D-Bibliothek für Einzelprozessoren für alle Architekturen gegeben.

Die Integration des Strömungslösers erfolgte in zwei Stufen. Zunächst wurde die sequentielle Variante integriert, später die parallelisierte, die einen völlig neuen Ansatz erforderte, s.u. Die Arbeiten wurden zusammen mit der DLR, GMD, HSB und NEC Europe realisiert.

Integration des sequentiellen FLOWer-Codes in COVISE

Aufgrund unterschiedlicher Programmiersprachen (COVISE in C++, FLOWer in FORTRAN77) ist eine komplexe Schnittstelle erforderlich, die von COVISE aufgerufen den Datenaustausch zwischen den beiden Programmen koordiniert. Außerdem bildet diese Schnitt-

stelle das Bindeglied zwischen dem Prozeßsteuerungsmodul PCM und FLOWer. Diese komplexe Schnittstelle, im weiteren FLOWer Modul (FM von HSB entwickelt) genannt, ruft erst, wie in Abb. 5 gezeigt, FLOWer als Unterprogramm auf.

FLOWer stellt die Schnittstellen zum FM bereit, die in enger Zusammenarbeit (Arbeits-Meetings und CSCW-Sitzungen) mit HSB abgestimmt und getestet wurden. Diese Schnittstellen gewährleisten, daß

- zu jedem Iterationsschritt die Konvergenzdaten Auftrieb, Widerstand und mittleres Residuum visualisiert werden können. Für die interaktive Steuerung von FLOWer sind diese Daten absolut notwendig, da sie dem Benutzer Aufschluß über die qualitative Entwicklung der Rechnung geben.
- die interaktiv durch das PCM geänderten Parameter für die Beeinflussung des Lösungsverfahrens unmittelbar in die nächste Iteration einfließen. Dies ist eine Grundvoraussetzung, daß die interaktive Steuerung der sehr schnellen FLOWer-Rechnung überhaupt zu realisieren ist.
- die Datenmenge zur Visualisierung auf die Geometrie-Oberfläche reduziert ist. Dies ist notwendig, um die Datenübertragung nicht zum Flaschenhals der interaktiven Arbeit werden zu lassen.

Der Datenaustausch erfolgt via Puffer, die mit den entsprechenden Daten gefüllt und anschließend an das FM weitergegeben werden. Die von FLOWer benötigten Daten für eine Rechnung werden vom PCM über diese Puffer verschickt aber nur dann berücksichtigt, wenn eine Modifikation des Datensatzes vorliegt. Die Konvergenzdaten können während der Rechnung am Bildschirm mit dem Public Domain Tool "XMGR" in einem Diagramm dargestellt werden. Die Daten der Geometrie-Oberfläche werden mit dem SURVIS Modul (DA) visualisiert.

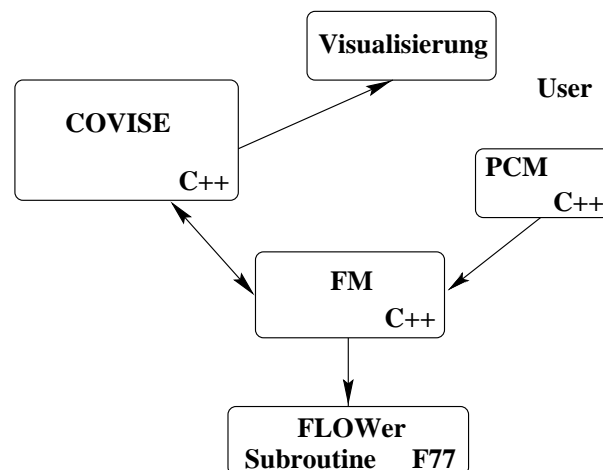


Abb. 5: Integration des sequentiellen FLOWer

Bei der Integration des sequentiellen FLOWer in COVISE wird die gesamte Prozeßkette, von

COVISE bis FLOWer auf einer Architektur zusammengelinkt. Diese Variante ist nicht auf den parallelen FLOWer-Code anwendbar, da die Parallelisierung auf dem Host-Node-Programmiermodell beruht und über MPI-Skripte (z.B. „mpirun“) gestartet wird. Eine Kopplung zum FM wie für den sequentiellen Code ist damit nicht möglich. Eine weitere Schwierigkeit besteht im ausgesprochen hohen Speicherbedarf der einzelnen Module. Auf „normal“ ausgestatteten Workstations bleibt nur noch wenig Platz für die FLOWer Rechnung bzw. das FLOWer-Modul. Industrierelevante Anwendungen können nicht mehr gerechnet werden. Weiterhin ist der Betrieb eines solchen (Gesamt-) Moduls auf einem Superrechner nicht sinnvoll, da meist die Graphikbibliotheken auf diesen Architekturen fehlen. Während eines zweitägigen Workshops mit allen Partnern (außer DA) wurde das weitere Vorgehen für die Integration des parallelen FLOWer festgelegt [12].

Integration paralleler FLOWer in COVISE

Aufgrund der oben beschriebenen Problematik wurde nach einem vollständig neuen Integrationsansatz gesucht. Im Vordergrund stand die lose Kopplung vom FM und dem parallelen FLOWer. Dies bedeutete, daß FLOWer völlig losgelöst von COVISE bzw. dem FM gestartet werden kann, trotzdem aber ein Datenaustausch für die interaktive FLOWer-Benutzung möglich ist.

Vom RUS wurde ein Konzept für die lose Kopplung auf der Basis von IP-Socket-Kommunikation vorgeschlagen, für das auf Erfahrungen aus vergangenen Projekten zurückgegriffen werden konnte [13]. Hierdurch wurde ein zusätzlicher An-/Abkoppelmechanismus möglich, was zu einem vollkommen neuen EFENDA-Szenario führte. In diesem Szenario startet der Anwender FLOWer auf einem beliebigen Superrechner, entweder von COVISE oder direkt als Batchjob. Während der gesamten Laufzeit kann er sich von der lokalen Workstation an den Superrechner ankoppeln und die aktuellen Daten zur Visualisierung holen, gegebenenfalls den Prozeß steuern und wieder abkoppeln. Vor allem für industrielle Langzeitjobs ist dies ein äußerst interessantes Szenario.

Die Kommunikation zwischen dem FM und dem parallelen FLOWer übernimmt ein Kopplungsinterface basierend auf IP-Sockets. Das von der GMD entwickelte Interface stellt die Routinen für das Versenden der Nachrichten von FLOWer nach FM und umgekehrt, sowie den Startmechanismus für ein Programm auf einem entfernten Rechner zur Verfügung. Dieses Kopplungsinterface basiert auf der DTM-Bibliothek (NCSA) [14].

In Abb. 6 ist die gesamte Prozeßkette dargestellt. Der wesentliche Unterschied zur Integration des sequentiellen Codes liegt in der losen Kopplung, d.h. FLOWer ist nicht mehr mit COVISE gelinkt. Über das Kopplungsinterface, direkt von COVISE aus, kann der parallele Code gestartet werden. Die Visualisierungsdaten und Prozeßsteuerungsdaten werden über vorher definierte IP-Sockets ausgetauscht. Durch einen spezifizierten Aufruf des Kopplungsinterfaces direkt in der FLOWer-Iterationsschleife wird überprüft, ob eine Kopplung zur COVISE-Schale vorliegt. Ist dies der Fall, werden die durch globale CLIC-3D Kommunikation erhaltenen Konvergenzdaten zu jeder Iteration verschickt.

Aufgrund gravierender Geschwindigkeitsunterschiede der verschiedenen Module waren Synchronisationsarbeiten notwendig. Beispielsweise müssen sämtliche FLOWer-Prozesse bei Inputänderung vor dem nächsten Zeitschritt angehalten werden. Weiter können die vom Parallelrechner sehr schnell produzierten Oberflächendaten nur zu festgelegten Zeitschritten übermittelt werden.

Die Graphikdaten für die Visualisierung mit SURVIS mußten als Charakter-Daten übertragen werden, was wesentlich langsamer ist als eine Kommunikation von Reals oder Integern. Nachdem dieser Punkt im industriellen Anwendertest bemängelt wurde, erfolgte die bereits zu Beginn des Projektes favorisierte Datenübertragung an das COVISE-FLOWer Modul.

Der parallele FLOWer kann sowohl im homogenen/heterogenen Rechnerverbund als auch auf Hochleistungsrechnern benutzt werden. Allerdings ist die zweite Variante aufgrund der höheren Kommunikationsleistung deutlich schneller, wie Tests von HSB und GMD (St. Augustin) zeigten [16].

Zukünftige parallele FLOWer Versionen werden auf der hostfreien CLIC-3D basieren. Hierdurch werden geringfügige Änderungen an einigen der hier entwickelten Schnittstellen notwendig.

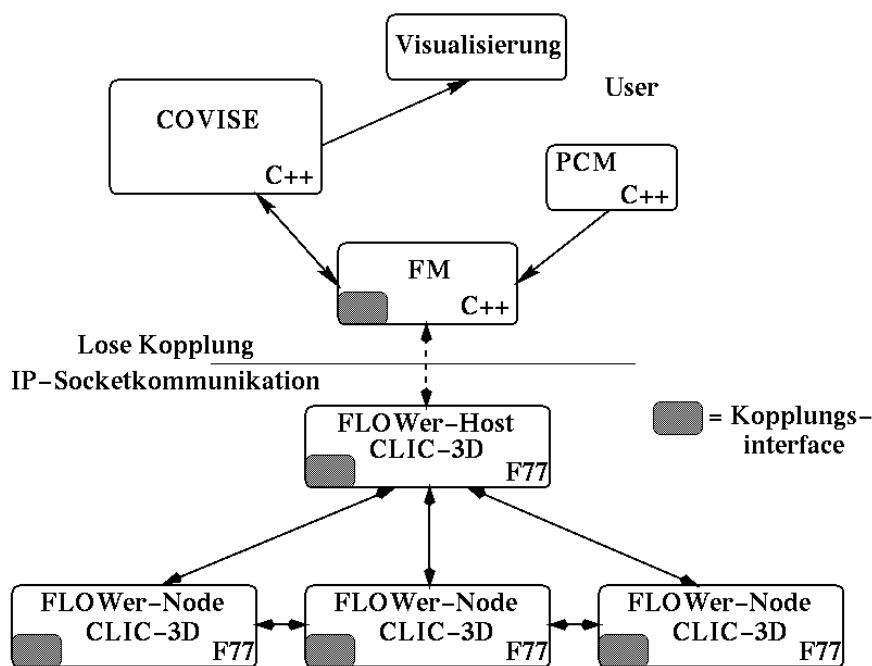


Abb. 6: Integration des parallelen FLOWer

3.4 Validierung der Funktionalitäten von MegaCads und FLOWer, WP4

Das durchgeführte Anwendungsszenario für den Einsatz der in EFENDA entwickelten Methoden ist aus dem typischen Vorgehen bei der Durchführung einer Computersimulation von aerodynamischen Strömungen um mehr oder weniger vollständige Flugzeugkonfigurationen abgeleitet. Diese Simulation umfaßt die Datenaufbereitung, d.h. die Diskretisierung (Netzgenerierung) des Strömungsfeldes einschließlich des umströmten Körpers, die Lösung der gekoppelten Differentialgleichungen zur Beschreibung der Strömung und die graphische Auswertung der berechneten Strömung.

Das in diesem Abschnitt vorgestellte Validierungsergebnis, spiegelt lediglich die Erfahrungen des DLR wider. Ein Erfahrungsbericht aus DA-Anwendersicht kann [3] entnommen werden.

Die Validierung gliedert sich in einen reduzierten Testfall, der als typisches Lernbeispiel gilt und einen realen Anwenderfall. Der reduzierte Testfall bestand aus der Generierung eines kleinen zweidimensionalen Netzes um einen Flügelschnitt, ca. 400 Netzpunkte und anschließender Rechnung. Der reale Anwenderfall bestand aus der Visualisierung eines großen realistischen Netzes (generische Flugzeugkonfiguration bestehend aus Flügel, Rumpf, Triebwerksgondeln, Pylon und Heckleitwerken), ca. 2.2 Mio Punkte, in MegaCads, anschließender Rechnung und Analyse der 3D-Oberflächenergebnisse. Dafür wurde zunächst eine Netzwerkverbindung aller Partner über das B-WiN aufgebaut, mit Ausnahme der Verbindung DA-HSB, die über das Landesbreitbandnetz Bremen erfolgte. In der ersten Phase saß die DA-Anwenderin (Testperson), die weder mit COVISE noch mit einem der anderen der Module je gearbeitet hat, an einer SGI-Workstation der HSB. Für die zweite Phase wurde statt der Workstation ein PC-Arbeitsplatz von DA über NetMeeting und X/GL-Server einbezogen. Die Verbindung der Partner ist in Abb. 7 dargestellt. Über die Desktop-Conference-Software InPerson [8] standen die Partner aus Bremen, Braunschweig, Berlin, Bonn und Stuttgart zusätzlich in Verbindung.

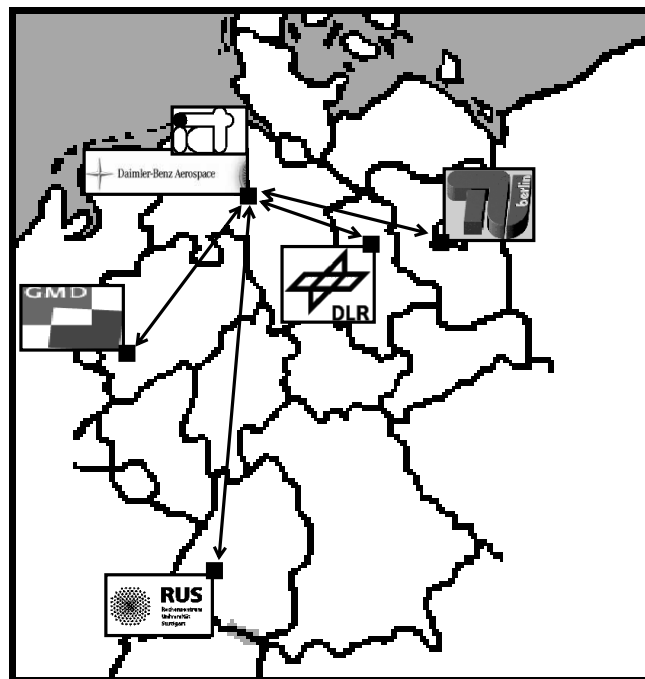


Abb. 7: Netzwerkverbindung für das EFENDA-Anwenderszenario

MegaCads

Der Einarbeitungszeit in COVISE inklusive der Module durch Teleteaching war gering. Das Lernbeispiel, ein einfaches zweidimensionales C-Netz um ein NACA-Profil wurde von der Testperson in Bremen generiert und anschließend geglättet. Obwohl die Testperson noch nie mit MegaCads gearbeitet hatte konnte sie den Netzgenerator bereits nach kurzer Zeit gut bedienen. Die dazu notwendige Unterweisung vom DLR Mitarbeiter in Braunschweig war

möglich durch die Mitverfolgung der Arbeitsschritte und der Inperson-Verbindung [8]. Praktisch gab es keine Einschränkung durch die räumliche Trennung der Teilnehmer.

Die kooperative MegaCads-Anwendung lief stabil. Allerdings führten nicht gesetzte MegaCads-Umgebungsvariablen oder eine Nicht-Ausführbarkeit, z. B. durch falsch gesetzte Rechte, zu einer Beendigung der COVISE-Sitzung bei allen Partnern.

Die Qualität der Bedienbarkeit und Performance der kooperativen MegaCads-Anwendung sind befriedigend. Das reale Netzbeispiel lief deutlich schneller als das Lernbeispiel, so daß ein allgemeiner Performanceverlust durch dritte Anwendungen oder eine erhöhte Netzbelastung angenommen werden muß. Eine genaue Lokalisierung war jedoch nicht möglich. Weiter ist anzunehmen, daß durch einen optimierten Kompilierungsvorgang eine spürbare Performance-Steigerung erreicht werden kann.

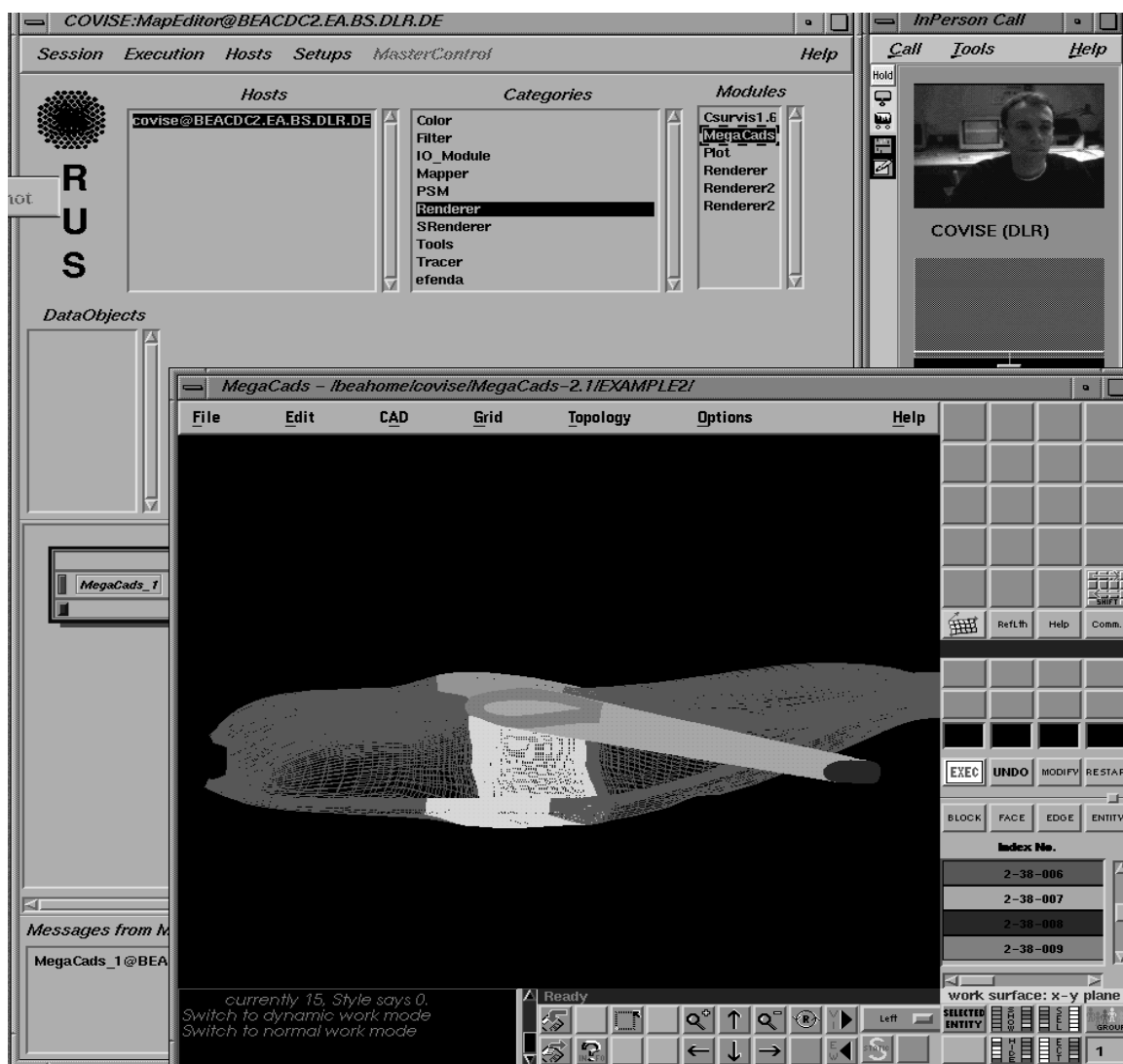


Abb. 8: Screen-Snapshot einer kooperativen MegaCads-Anwendung

Abb. 8 zeigt einen Screen-Snapshot eines realen Anwendungstestfalls mit MegaCads. Diese Ansicht stellt sich überlicherweise bei der kooperativen Arbeit ein: Im Hintergrund sichtbar, die multimediale Arbeitsumgebung COVISE mit dem integrierten MegaCads-Modul. Oben rechts im Bild, die Videobilder der InPerson Software und schließlich die graphische Bedienoberfläche von MegaCads.

FLOWer

Das DLR mußte keine Daten für dieses Szenario übertragen, so daß keine Arbeiten vor der kooperativen Sitzung geleistet werden mußten.

Der FLOWer-Code lief in der sequentiellen und parallelen Variante ohne Einschränkungen stabil. Die Optionen wie An- und Abkoppeln sowie Pause/Fortsetzen funktionierten reibungsfrei. Performance-Tests auf Hochleistungsrechnern und homogenen oder heterogenen Rechnerclustern zeigten, daß die Parallelrechnung auf einem Hochleistungsrechner dem verteilten Rechnen im B-WiN deutlich überlegen ist [16]. Probleme traten an der Schnittstelle SURVIS-COVISE auf: Nach einem Master-Slave-Wechsel war keine neue Ansicht der visualisierten Oberflächenergebnisse mehr möglich, so daß eine Diskussion der Strömungsergebnisse nicht mehr möglich war. Auch hier kam es zu Abstürzen der gesamten Session, wenn ein Partner nicht die richtige Konfiguration eingestellt hatte.

Eine mangelnde Performance der kooperativen FLOWer Anwendung fiel lediglich bei der Visualisierung der Oberflächendaten auf. Diese Wartezeit konnte noch während der Sitzung auf akzeptable 30 sec. reduziert werden.

Das Hinzuschalten von DA (PC-Arbeitsplatz statt Workstation) führte zu keinen wesentlichen Änderungen der oben gemachten Aussagen. Lediglich das MegaCads-Modul konnte nicht auf den PC umgelenkt werden. Die Fehlermeldung läßt allerdings den Schluß zu, daß die MegaCads-Sourcen zu einem falschen Zeitpunkt, nämlich vor dem Umlenken des Displays gesetzt wurden.

3.5 CLIC-3D Implementierung in FLOWer für heterogenen Rechnerverbund, WP5

Die Implementierung der Bibliothek brachte keine notwendigen Änderungen im Strömungslöser-Code mit sich. Das DLR mußte keine Arbeiten in diesem Arbeitspaket erledigen. Durch den besonderen Ansatz, der für die Integration des parallelen FLOWer-Codes gefunden werden mußte und die damit verbundenen zusätzlichen Arbeiten entstand im Arbeitspaket WP4 einerhöhter Bedarf. Dieser konnte durch die Verschiebung der 0.1 Personjahre aus WP5 nach WP4 ausgeglichen werden.

3.6 Schnittstellendefinition für interaktive Prozeßsteuerung in COVISE, WP6

MegaCads

In enger Absprache mit RUS wurde zunächst die Art der Schnittstelle definiert. Für MegaCads in COVISE wurde der Typ Renderer gewählt. Die Art des Datenaustauschs über das Versenden

von Messages wurde bereits oben beschrieben. Für die Behandlung des Datentransfers wurden zwei neue MegaCads-Module eingerichtet, die die Character-Messages auf Masterseite generieren und auf Slave-seite wieder entschlüsseln. Aufgrund der komplexen Datenstruktur in MegaCads mußten zahlreiche Änderungen im Code vorgenommen werden, um das korrekte Versenden der Character-Messages für alle Funktionalitäten zu gewährleisten.

FLOWer

Das DLR erarbeitete gemeinsam mit TUB, RUS und HSB das Konzept des Prozeßsteuerungsmoduls (PCM). Grundlage der Arbeiten bildete das Input Datafile inp.dat, das alle Parameter zur Steuerung des FLOWer Codes enthält. Für eine verbesserte Bedienbarkeit des PCM wurde eine Gruppierung der Parameter vorgenommen, wie sie derzeit in FLOWer noch nicht existiert. Sie unterscheidet die zu setzenden Parameter in:

- Allgemeine Kontrolle,
- Turbulenz,
- Transition,
- Multigrid,
- Rechenetz und
- Netzblöcke

Auch dem ungeübten FLOWer-Nutzer ist somit die Möglichkeit gegeben, die von ihm zu modifizierenden Werte gezielt anzusteuern und alle übrigen unberührt zu lassen. Desweiteren sind für alle Parameter Grenzen vergeben worden, die eine Überprüfung der Richtigkeit des gesetzten Wertes zuläßt.

Im FLOWer-Code sind Änderungen durchgeführt worden, die die permanente Änderung der Steuerungsparameter zulassen. Dazu wird zu jeder Iteration eine Routine des FLOWer Moduls aufgerufen, um die Input-Puffer zu empfangen. Falls die Input-Daten über das PCM geändert worden sind, werden sie in FLOWer aktiviert.

4.0 Wissenschaftlicher und technischer Stand vor Beginn des Projektes

Das DLR ist die einzige Großforschungseinrichtung Deutschlands auf dem Gebiet der Luft- und Raumfahrt. Es besitzt Erfahrungen und fundierte Kenntnisse in allen Bereichen des Entwurfs und der Analyse von Flugzeugen und deren Komponenten. Das Institut für Entwurfsaerodynamik verfügt über weitreichende Kenntnisse auf dem Gebiet der angewandten Aerodynamik und Akustik und stellt somit ein Bindeglied zwischen Theorie und praktischer Anwendung dar.

Die Entwicklung und Anwendung numerischer Simulationswerkzeuge sind wesentliche Arbeitsgebiete des Instituts für Entwurfsaerodynamik. Das Projekt MEGAFLOW gehört dabei zu den Projekten, die die Erfahrungen der Mitarbeiter im Bereich Netzgenerierung und Strömungsberechnung intensiv vorantreiben. Da ein Großteil der Softwareentwicklungen im Haus durchgeführt wird, konnte das Institut mit seinem know-how bei der Integration von FLOWer und MegaCads in COVISE und natürlich bei ihrer Bedienung stark unterstützend bei den Partnern tätig werden.

Auf der anderen Seite hat das Institut nur wenig bis gar keine Erfahrungen auf dem Gebiet des kooperativen Arbeitens unter Benutzung geeigneter Bedienoberflächen wie z.B. COVISE. Insgesamt ist der Bereich multimediales Arbeiten noch unterrepräsentiert. Daraus resultierten unter anderem die infrastrukturellen Maßnahmen sowie die Einarbeitung in Desktop-Konferenz-Software und ihrer Konfiguration durch einen IP-Multicast-Routing-Daemonen.

5.0 Zusammenarbeit mit anderen Stellen

Neben dem Deutschen Zentrum für Luft- und Raumfahrt (DLR) nahmen folgende Projektpartner an dem Projekt EFENDA teil:

- 1) Daimler-Benz Aerospace Bremen (DA)
- 2) Institut für Aerospace-Technology der Hochschule Bremen (HSB)
- 3) Rechenzentrum der Universität Stuttgart (RUS)
- 4) Die Gesellschaft für Mathematik und Datenverarbeitung (GMD)
- 5) Hermann-Föttinger-Institut der TUB (HFI)

Alle EFENDA Partner wurden assoziierte Partner des MEGAFLOW-Konsortiums. Dadurch stehen den Partnern die neuesten Versionen von MegaCads und FLOWer zur Verfügung.

Die aus Sicht des DLR gute Zusammenarbeit wird im wesentlichen auf die konsequente Ausnutzung moderner Kommunikationsmittel zurückgeführt. Obwohl das Kontaktieren der Partner und das schnelle Verteilen von Informationen, es wurde beispielsweise ein spezieller EFENDA-Mail-Server eingerichtet, das dezentrale Zusammenarbeiten stark unterstützte, konnten die Videokonferenzen persönliche Meetings nicht vollständig ersetzen.

Zusätzliche Arbeiten, vor allem die Weiterentwicklung der Kommunikationsbibliothek CLIC-3D, wurden vom Unternehmen NEC durchgeführt. Hintergrund war der Wechsel eines in das Projekt involvierten GMD-Mitarbeiters zu NEC Europe in Bonn.

6.0 Erzielte Ergebnisse

Die Programmsysteme MegaCads und FLOWer wurden erfolgreich in die multimediale kooperative Benutzerumgebung COVISE integriert. Kosten und Zeitplanung wurden eingehalten. Geringfügige Verzögerungen in der Zeitplanung konnten während des Projektes wieder ausgeglichen werden. Das DLR hat alle Arbeitspakete des EFENDA-Projektes erfolgreich abgeschlossen.

Infrastruktur

Das DLR verfügt heute über die notwendige Hardware, Workstation und Breitbandanschluß, die es ermöglicht, auf multimedialer Ebene zu kommunizieren und zu arbeiten. Durch zahlreiche virtuelle Gruppenmeetings (zwei bis fünf Personen) konnte diese Art des Arbeitens bewertet werden. Obwohl insgesamt ein positiver Eindruck besteht, ist zu bemerken, daß die Audio- und Videoübertragung über das B-Win niemals bei allen Sitzungen mit mehr als zwei Teilnehmern, unabhängig von der Tageszeit, über einen längeren Verlauf störungsfrei funktionierte. Das praktische Vorgehen bestand darin, die Videoübertragung zu stoppen und sich auf eine Audioübertragung mit Whiteboard zu beschränken. Es war nicht festzustellen, wodurch die Verzögerungen ausgelöst wurden. Das Institut wird trotzdem versuchen, diese effiziente Art der Kommunikation verstärkt zu nutzen.

MegaCads-Integration

Der Netzgenerator MegaCads ist vollständig in die COVISE Umgebung integriert worden. Die Verfügbarmachung sämtlicher in MegaCads-2.1 vorkommender Funktionalitäten ermöglicht den Teilnehmern das komplette Netz kooperativ zu generieren. Die Arbeitsweise des Masters ist dabei hervorragend von allen Teilnehmern mitzuverfolgen. Durch die realisierte kooperative GUI-Gestaltung ist vollständig der Eindruck einer gemeinsamen Arbeitssitzung gegeben. Alle Teilnehmer können die einzelnen Arbeitsschritte des Masters über das MegaCads-Bedienoberfläche mitverfolgen: Die Entity-Liste wird gemeinsam bewegt, Popup-Windows gehen bei allen Partnern auf und die Menüauswahl zur Selektion der gewünschten Funktion, z.B. Konstruktion eines Splines oder Glätten eines zweidimensionalen Netzes, wird gemeinschaftlich durchgeblättert und ausgeführt. Die Bewegung des Mauszeigers über die interaktive Bedienoberfläche ist dabei ein wesentliches Element bei der kooperativen Arbeit. Die Ansichten des augenblicklich bearbeiteten Netzes werden natürlich bei allen Partnern synchronisiert.

Der in COVISE erfolgreich integrierte Netzgenerator MegaCads ist, ohne Einschränkung seiner Funktionalitäten, aufgrund des gewählten COVISE-Modultyps „Renderer“ nicht in die Prozeßkette Preprocessing-Rechnung-Postprocessing integrierbar (vgl. Abb. 1). Der Grund liegt in der Definition des COVISE Renderer-Moduls. Dies besitzt eine Schnittstelle für Inputdaten, jedoch keine für den Output. Das bedeutet, daß MegaCads zwar kooperativ unter COVISE bedient werden kann, jedoch keine Daten an das Folgemodul FLOWer weitergeben werden können. Eine Umstellung des Modultyps für die Integration ist dann angebracht, wenn MegaCads in der Lage ist, sämtliche für die Rechnung benötigten Daten zur Verfügung zu stellen - insbesondere die Topologiedaten. Dieser Entwicklungsstand lag bei der integrierten Version 2.1 noch nicht vor.

FLOWer-Integration

FLOWer ist in seiner sequentiellen und parallelen Variante in die multimediale Oberfläche COVISE integriert worden. Dadurch ist folgendes Szenario möglich geworden: Ein CFD-Anwender beispielsweise in Bremen startet COVISE und das darin enthaltene FLOWer-Modul. Nach Ankoppeln an einem High Performance Computer (HPC) in Frankfurt stellt er Konvergenzprobleme bei der Rechnung fest. Mit Hilfe seines Videokonferenzsystems nimmt er über das schnelle B-WiN kontakt zu dem gewünschten Experten in Braunschweig auf und bezieht diesen in seine derzeitige am Bildschirm befindliche Applikation mit ein. Die FLOWer-Rechnung wird vom Master angehalten und der Experte unterstützt bei der Auswahl geeigneter Parameter oder übernimmt selbst die Master-Rolle. Die Rechnung wird fortgesetzt und liefert nun ein brauchbares Ergebnis, das zusammen mit dem zusätzlich einbezogenen Flugzeughersteller in Hamburg diskutiert werden kann. Damit ist ein Maximum an Geschwindigkeit für Informationsverbreitung und -verarbeitung erreicht worden.

Ein großer Fortschritt ist bei der Variabilität der Rechenmaschinen erzielt worden. FLOWer kann sequentiell und parallel auf homogenen Parallelrechnern und heterogenen Rechnerclustern ausgeführt werden. Mit der im Projekt von DLR, GMD, HSB und RUS gemeinsam erarbeiteten Lösung für die Integration des parallelen FLOWer-Codes wurden mehrfach Szenarien mit verschiedenen Workstations und Rechnern durchgeführt. Dabei kontrollierte z.B. ein Anwender auf einer Workstation der Hochschule Bremen den FLOWer-Simulationslauf auf einer SGI-Origin der GMD, St. Augustin. Die Übertragung der Oberflächendaten über das B-WiN war unproblematisch, Verzögerungen traten lediglich durch einen langsamen Firewall Rechner (PC) bei der Hochschule auf.

Durch das vom HFI entwickelte und mit dem DLR abgestimmte PCM-Modul ist die Eingabe der FLOWer spezifischen Parameter stark vereinfacht worden. Durch die neu eingeführte Kontrolle der eingegebenen Parameter, jedem Parameter wurde ein spezifischer Gültigkeitsbereich zugewiesen, können benutzerabhängige Fehler schneller identifiziert werden.

Gerade die Vielfalt der für FLOWer zu setzenden Parameter und der mitunter schnellen Wirkung auf das Konvergenzverhalten der Strömungsberechnung macht das kooperative Arbeiten mit FLOWer über PCM und SURVIS zu einer nützlichen Anwendung, die der herkömmlichen Vorgehensweise deutlich überlegen ist.

Das Konzept für die FLOWer Integration ist äußerst flexibel und somit nicht ausschließlich an die COVISE-Umgebung gebunden. Daher ist es naheliegend, daß es für zukünftige Projekte, wie z.B. AMANDA (Nutzung und Kopplung unterschiedlicher Simulationsprogramme unter einer einheitlichen Benutzeroberfläche), eine ausgezeichnete Basis bilden kann.

7.0 Voraussichtlicher Nutzen

Die Verfügbarmachung sämtlicher in MegaCads-2.1 vorkommender Funktionalitäten ermöglicht den Teilnehmern das vollständige Netz kooperativ zu generieren. Allein aus praktischen Überlegungen wird es hauptsächlich zur Nutzung einzelner Funktionen kommen.

Die Bedeutung einer kooperativen im Vergleich zur herkömmlichen Vorgehensweise mit MegaCads hängt von der Zieldefinition ab. Wie aus der Validierungsarbeit hervorgeht, kann ein ungeübter Nutzer mit der kooperativen Arbeitsweise in sehr kurzer Zeit den Umgang mit dem Netzgenerator erlernen. Dabei verbessert eine konsequent ausgenutzte Schüler-Lehrer Situation mit einer beliebig großen Gruppe von Teilnehmern das Verhältnis von Nutzen zu Aufwand. Für die gewinnbringende Nutzung des Moduls MegaCads als Netzgenerator und nicht als Visualisierungstool für sehr große generierte Netze gilt im wesentlichen das gleiche, mit dem Unterschied, das die Vorbereitungsphase mehr Zeit in Anspruch nimmt. Ein kooperatives Arbeiten zweier Experten gilt jedoch als eher unwahrscheinlich, da die Schwierigkeiten der Netzgenerierung eher topologischer als konstruktiver Natur sind. Für ersteres dürfte bereits eine 'Whiteboard-Funktionalität' zur Problemdiskussion ausreichen.

Die für MegaCads gemachten Aussagen gelten nicht für das kooperative Arbeiten mit dem Strömungslöser FLOWer. Da es sich im wesentlichen um ein nicht-interaktives Programm handelt, ist es während der Rechnung im allgemeinen nicht notwendig, daß alle Partner bei einer Sitzung anwesend sind. Sie können temporär in die Diskussion mit einbezogen werden, wenn es zum Beispiel um die Beseitigung von Konvergenzproblemen oder um die Interpretation erzielter Strömungsergebnisse geht. Das schnelle „Hinzuschalten“ von Experten macht die kooperative Arbeit somit äußerst interessant.

Die Kopplung eines entfernten Parallelrechners mit einer Graphikworkstation zur Visualisierung laufend produzierter Strömungsergebnisse, erleichtert die Benutzung von FLOWer sowohl für geübte als auch ungeübte Nutzer. Es soll in Zukunft versucht werden, verstärkt von der neuen Technologie Gebrauch zu machen. Je nach Verfügbarkeit kann die parallele Anwendung auf homogenen und heterogenen Rechnerclustern oder auf Hochleistungsrechnern stattfinden.

8.0 Fortschritt von anderen Stellen

In den letzten Jahren konzentrierten sich aufgrund beschränkter Rechnerleistungen und eingeschränkter informationstechnologischer Fähigkeiten die bisherigen Anstrengungen auf monodisziplinäre Einzelprobleme. Der neue Trend geht jedoch in ganzheitliche, interdisziplinäre Entwurfs- und Optimierungsstrategien, wie im Projekt EFENDA verwirklicht. Dem Institut sind folgende Fortschritte von anderen Stellen bekannt:

- SUPEA: Entwicklung eines interaktiven „numerischen“ Prüfstandes für Strömungsanalysen im Antrieb, einschließlich Verbrennung und Schadstoffbildung.
- SPOCK: Schnelle Produktentwicklung durch optimierte computerunterstützte Kommunikation.
- AMANDA: Zusammenfassung von Simulationsprogrammen zu einem interdisziplinären Softwaresystem unter einer gemeinsamen Benutzeroberfläche.

Ferner ist bekannt, daß Probleme des Hochleistungsrechnens in unterschiedlichsten Forschungsverbänden behandelt werden. Ein Beispiel ist der Bayerische Forschungsverbund für technisch-wissenschaftliches Hochleistungsrechnen (FORTWIHR), in dem in interdisziplinärer Zusammenarbeit zwischen Ingenieurwissenschaften, angewandter Mathematik und Informatik Programmsysteme für Hochleistungsrechner entwickelt und für praktische Anwendungen nutzbar gemacht werden.

9.0 Veröffentlichungen

Von Seiten des DLR sind keine Veröffentlichungen in Bezug auf EFENDA geplant. Vom Projektleiter geplante Veröffentlichungen unter Beteiligung des DLR sind dem Schlußbericht von DA zu entnehmen.

10.0 Literatur

- [1] F. Arnold, K. Becker, Effiziente Flugzeugentwicklung durch Nutzung von Datenautobahnen (EFENDA) - Beschreibung des BMBF-Projektes - 30.10.1995
- [2] F. Arnold, F., Bleecke, R. Ziegler, G. Ostwald, H. Schwichtenberg, A. Oetken, H. Nebel, J. Beutner, H. Cornelius. EFENDA - Verteilte Entwicklung am Beispiel des Flugzeugentwurfs - Zwischenbericht BMBF-Verbundprojekt 01 BN 606/9, Feb.97.
- [3] P. Aumann, Erfahrungsbericht über die Anwendung des COVISE-Systems zur Durchführung industrieller Entwurfsarbeiten bei Dasa Airbus. EF-016/98. Dasa Airbus Bremen. März 1998.
- [4] W. Bartelheimer, Projektbeschreibung MEGAFLOW, DLR-IB 129-96/8, Braunschweig 1996
- [5] W. Bartelheimer, H. Bleecke, B. Eisfeld, J. Liser, R. Heinrich, N. Kroll, M. Kuntz, E. Monsen, J. Raddatz. Installation and User Handbook for the Projeet FLOWer, QS-FLOWer-3008, DLR 21.3.1996.
- [6] O. Brodersen, M. Hepperle, A. Ronzheimer, C.-C. Rossow, B. Schöning. The Parametric Grid Generation System MegaCads. Proc. of the 5th Intern. Conference on Numerical Grid Generation in Computational Field Simulations. NSF Mississippi 1996.
- [7] O. Brodersen, A. Ronzheimer, R. Ziegler, T. Kunert, M. Hepperle, J. Wild, R. Heinrich. Aerodynamic Applications Using MegaCads. Proc. of the 6th Intern. Conference on Numerical Grid Generation in Computational Field Simulations. London GB 1998.
- [8] InPerson User Guide, Silicon Graphics, 1996
- [9] R. Lang, U. Lang, H. Nebel, D. Rainer, D. Rantzau, A. Wierse, U. Wössner. COVISE Version 3.0a. RUS. Sept. 96.
- [10] MegaCads-Online. <http://www.bs.dlr.de/~ea1v/HelpMain.html>
- [11] H. Nebel. Integration von MegaCads in COWESTA - Realisierung als Modultyp GENERIC und Realisierung eines Datenports zum FLOWer Modul, EFENDA Bericht EN-WP 4-05, RUS, Feb. 1998.
- [12] H. Nebel. Protokoll EFENDA-Arbeitstreffen. RUS. Stuttgart 7.5.1997
- [13] H. Nebel, U. Lang, D. Rantzau, A. Wierse. EFENDA Bericht. Integration des parallelen FLOWer in COVISE. EN WP 5-01. RUS 16.7.97.
- [14] NCSA Data Transfer Mechanism, Programming Manual, National Center of University of Illinois, Feb. 1992

- [15] J. Raddatz, P. Funk, N. Kroll. Architectural Design Handbook for the Project FLOW-
Wer. Doc.Nr. MEGAFLOW-1002, DLR 26.9.1996.
- [16] H. Schwichtenberg, Verteilte Entwicklung am Beispiel des Flugzeugentwurfs
(EFENDA), Teilvorhaben: Kommunikationsbibliotheken, Abschlußbericht, Insti-
tute für Algorithmen und Wissenschaftliches Rechnen, St. Augustin 1998