Projekt FORAUS

Anforderungsdefinition

3. Projektsemester
WS93/94

Fachbereich Informatik

Projektbetreuung:
Prof. Dr. Bernd Krieg-Brückner, Michael Fröhlich, Mattias Werner

6. Ein-/Ausgabeverhalten

Ein wichtiger Aspekt bei einer Dokumentenverarbeitung ist die Druckausgabe. Sie ist in den meisten Fällen das eigentliche Ziel der gesamten Bearbeitung. Es muß versucht werden, hier vernünftige Resultate zu erzielen.

Ein weiterer Punkt ist das Laden und Speichern von Dokumenten. Welche Funktionalitäten hier bereitgestellt werden müssen, soll im Folgenden kurz erläutert werden.

Dateifunktionen

Als Dateifunktionen müssen mindestens folgende zur Verfügung stehen:

- Datei laden
- Datei speichern
- Datei alle n-Minuten speichern

Alle diese Aktionen werden durch Dialoge gesteuert. Inwieweit der Benutzer durch Auswahllisten (File-Browser) unterstützt wird, hängt vom Betriebssystem ab, welches diese Routinen vielleicht schon zur Verfügung stellt.

Druckfunktionen

Über ein Dialog kann das Dokument auf einen Drucker ausgegeben werden. Im Dialog ist hierzu einstellbar:

- alle Seiten drucken
- von Seite bis Seite drucken
- Anzahl der Kopien

Das Erscheinungsbild des Dokumentes auf dem Papier hat dem der WYSIWYG-Sicht annähernd zu entsprechen. Als Ausgabeformat ist Postscript zu verwenden.

kontextbezogene Formate

Die Formate müssen kontextbezogen definierbar sein, z. B. ein Format „wichtiges Wort“ hat im Kontext eines Formates „Fließtext“ eine andere Formatierung zur Folge als in „Fußnote“. In diesem Beispiel dürften nicht alle, sondern nur bestimmte Formatierungsmerkmale geändert werden (Schriftschnitt = kursiv; Schriftart und Schriftgrad bleiben erhalten).

Kopf- und Fußzeile

Es müssen ebenfalls Informationen einzugeben sein, was in der Kopf- und Fußzeile erscheinen soll. Dies hat nicht in den Präsentationsregeln zu erfolgen, sondern interaktiv durch den Benutzer bei der Dokumentenbearbeitung. Neben festem Text sollte es möglich sein, einen Querverweis in der Kopf- bzw. Fußzeile zu hinterlegen, der z. B. die Überschrift des aktuellen Kapitels bzw. die Beschreibung (Überschrift) eines beliebigen Formates enthält.

Fußnoten

Die Präsentationsregeln müssen die Realisierung von Fußnoten ermöglichen. Nach der Zuweisung eines Formates „Fußnote“ (z. B.) erscheint an die Stelle des ursprünglichen Textes eine für Fußnoten übliche hochgestellte Zahl. Sie ist logisch mit dem vorhergehenden Objekt im Text verbunden. Die Zahl richtet sich nach der Anzahl der bereits eingegebenen Fußnoten. Der so ausgezeichnete Text hat nun am Ende der Seite unter einem Abtrennungsstrich und führend mit der Referenzzahl zu erscheinen. Verschiebt sich das logisch verknüpfte Objekt, auf eine andere Seite, so hat auch der Fußnotentext auf dieser neuen Seite zu erscheinen.

Silbentrennung (wünschenswert)

Oben wurde bereits beschrieben, daß ein Zeilenumbruch beim Erreichen der Zeilenbreite zu erfolgen hat. Um einen harmonischeren Umbruch zu erhalten, sollte nicht nur am Ende, sondern auch innerhalb eines Wortes getrennt werden. Diese Silbentrennung muß korrekt und möglichst optimal erfolgen. Wünschenswert wäre eine global optimierte Silbentrennung. Die Silbentrennung muß abschaltbar sein.

Seitenumbruch (wünschenswert)

Beim Erreichen des Seitenendes muß der Text auf die nachfolgende Seite umgebrochen werden. Die Regeln des Zeilenumbruches sind hier ebenfalls anzuwenden. Der Seitenumbruch muß das Vorhandensein von Fußnoten entsprechend berücksichtigen.

5. Dokument-Formatierung

Die Formatierung des Dokumentes geschieht allein durch das System, auf der Grundlage der Präsentationsregeln. Im Abschnitt „Dokumenten-Eingabe“ wurde beschrieben, daß Objekten Formate zugewiesen werden. Es kann kein Objekt ohne eine solches Formatattribut vorkommen.

Formatierungsmerkmale

In den Präsentationsregeln ist beschrieben, wie Objekte eines bestimmten Formates zu formatieren sind. Es müssen mindestens die folgenden Formatierungsmerkmale in den Präsentationsregeln eingegeben werden können, die dann das Formatierungsverhalten der Formate bestimmt:

a) Schriftart
   Mindestens die Schriften Times und Helvetica in Proportionalschrift.

b) Schriftgröße
   Die Schriftgrößen sollten zwischen 5pt und 72pt (Punkt) einstellbar sein.

c) Schriftschnitte
   Es müssen mindestens normal, fett, kursiv und unterstrichen möglich sein.

d) Ausrichtung
   Die Einstellung der Ausrichtung für Absätze muß mindestens linksbündig, rechtsbündig, zentriert und Blocksatz ermöglichen.

e) Absatzeinzug
   Der linke Einzug am Anfang eines Absatzes muß in mm frei einstellbar sein.

f) Zeilendurchschuß
   Der Zeilendurchschuß muß ebenfalls frei einstellbar sein. Die Einheit ist Punkt (pt).

g) Absatzabstand
   Der Abstand über und unter einem Absatz muß in mm einstellbar sein.

h) Führende (Sonder-) Zeichen
   Es muß möglich sein Formate zu definieren, die z.B. eine Aufzählung realisieren. Diese zeichnet sich entweder durch einen führenden Punkt oder einer fortlaufenden Numerierung aus. Es ergibt sich die Forderung, eine Numerierung abschnittsweise vorzunehmen. Der Benutzer muß also angeben können, daß diese ab dem nächsten Element des Formates bei einem Startwert neu beginnen soll.

Layout-Angaben

Neben diesen Schriftfestlegungen und Absatzangaben muß es möglich sein, Angaben für das gesamte Dokument einzugeben (Layout). Hierzu zählen Positionsangaben für Kopf- und Fußzeilen, Position der Paginierung und Angaben zur Zeilenbreite. Eine Zeile darf die Zeilenbreite nicht überschreiten und muß u. U. dort umgebrochen werden. Ein Umbruch darf nicht an Stellen erfolgen, an denen geschützte Leerzeichen eingegeben wurden (siehe auch Abschnitt „Dokumenten-Eingabe“).

Anzeige der Restverarbeitungszeit bei Funktionsausführung (wünschenswert)

Löst der Benutzer eine komplexe Funktion aus, die eine hohe Verarbeitungszeit erfordert, so wird ihm angezeigt, wie lange die Ausführung dieser Funktion noch dauert.

Speicherung des Arbeitsbereichs

Wird ein Dokument geladen, so werden genau die Fenster, die beim letzten Beenden des Dokumentes offen waren, geöffnet. D. h. der Arbeitsbereich des Benutzers bezogen auf das Dokument ändert sich nicht. Desweiteren wird der Text-Cursor immer an die zuletzt gewählte Stelle gesetzt.

Allgemeines Systemverhalten

selektierte Objekte kennzeichnen

Wenn der Benutzer Objekte selektiert, wie z. B. beim Markieren mit Block-Funktionen, so müssen sich diese ausgewählten Objekte in geeigneter Form von ihrer Umgebung unterscheiden.

passive Hilfe

Dem Benutzer sollte es möglich sein, eine funktionsbezogene Hilfe aufzurufen.

automatisches Scrollen bei Berührung der Fenstergrenze

Kommt der Benutzer beim Markieren des Textes an die Fenstergrenze, so muß das System automatisch „weiter-scrollen“ und die folgenden Textzeilen anzeigen.

Bestätigung kritischer Funktionen

Wird eine kritische Funktion wie Beenden oder Löschen aufgerufen, deren Ausführung zu einem Informationsverlust führen könnte, dann muß das System beim Benutzer rückfragen, ob die Funktion tatsächlich ausgeführt werden soll.

Versionsverwaltung (wünschenswert)

Es muß dem Benutzer möglich sein, alle Zustände in denen sich das Dokument während der Erstellung befunden hat, wieder herzustellen. Das Dokument befindet sich in einem neuen Zustand (Version), wenn es nicht mehr das „aktuelle Dokument“ ist, d. h. wenn ein neues Dokument geladen bzw. erstellt wird oder wenn das Programm beendet wird.

Größenanpassung des Textcursor an Buchstaben (wünschenswert)

Der Textcursor muß sich automatisch der Art und Größe des Buchstabens, auf dem er gerade steht, anpassen. D. h. läuft der Cursor über eine Zeile die in einem 24 Punkt Font gesetzt ist, so muß er auch 24 Punkt groß sein. Ist die Zeile in einem kursiven Font gesetzt, so muß auch der Cursor schräg gestellt sein.

automatisches Rückspeichern von Texten (wünschenswert)

Der Benutzer sollte die Möglichkeit haben, ein Zeitintervall anzugeben, nach dem das System das aktuelle Dokument während der Bearbeitung automatisch rückspeichert. Diese Funktion sollte optional sein.

Anzeige möglicher UNDO/REDO-Schritte (wünschenswert)

Die jeweils möglichen UNDO- und REDO-Schritte sollten angezeigt werden. Hat man z. B. gerade ein Wort gelöscht, so müßte dem Benutzer ein „UNDO - löschen“ zur Verfügung stehen. Führt er dies aus, müßte ihm ein „REDO - löschen“ zur Verfügung stehen.

ASCII-Sicht

Die ASCII-Sicht stellt den reinen Text des Dokumentes ohne logische Zuordnung, Kopfzeile, Fußzeile, Grafiken usw. dar. Die Funktionalität, die dem Benutzer hier zur Verfügung steht, sollte bis auf zwei Ausnahmen dieselbe sein wie bei WYSIWYG.

- Das Setzen interner Querverweise ist nicht möglich.
- Die Block-Funktion „logisches Element zuweisen“ ist nicht möglich.

Sollte es beim Markieren von Textstellen zu Konflikten bezüglich ihrer logischen Zuordnung kommen, die obwohl sie in dieser Sicht nicht dargestellt wird intern trotzdem vorhanden ist, so muß das System diese Konflikte erkennen und lösen. D. h. bei Markierungen über logische Grenzen hinweg, werden nur vollständig ausgewählte logische Elemente tatsächlich markiert.

PostScript-Sicht

Diese Sicht stellt das Dokument in der PostScript-Notation dar. Es handelt sich hierbei um eine reine Ausgabesicht, die den Text und die zugehörigen PostScript-Steuerzeichen zeigt. Es stehen dem Benutzer hier keine besonderen Funktionalitäten zur Verfügung, insbesondere darf es ihm nicht möglich sein, in dieser Sicht zu editieren

Struktursicht

Diese Sicht stellt die Struktur des Dokumentes dar. Es sollen alle logischen Elemente die laut der DTD auch im Inhaltsverzeichnis auftauchen als Objekte dargestellt werden, die baumartig in ihrer Hierarchie verknüpft sind. Es muß möglich sein, einzelne oder mehrere dieser Objekte zu markieren und dann auszuschneiden, einzufügen und zu kopieren, wobei sich die logische Zuordnung des Elementes aber nicht ändern darf.

Die Objekte sollen so dargestellt werden, daß man möglichst viel des zugehörigen Textes im Objekt lesen kann (z. B. einzeilig) und auch die logische Zuordnung ablesbar ist.

Allgemeines zu den Sichten:

WYSIWYG - Default

Die WYSIWYG-Sicht sollte die ‘Default-Sicht’ sein. Dies bedeutet, daß beim Erstellen eines neuen Dokumentes dieses zuerst automatisch WYSIWYG dargestellt wird.

Änderungen

Bei Änderungen in einem Fenster einer Sicht sollen alle anderen offenen Fenster aktualisiert werden.

gleiche Sichten

Es muß möglich sein mehrere gleichartige Sichten auf ein Dokument (z. B. an verschiedenen Stellen des Dokumentes) zu öffnen.

„Fenster-Anker“

Ein Fenster bleibt immer an derselben Textstelle im Dokument, selbst wenn vor dieser Stelle in einem weiteren Fenster etwas hinzugefügt wird.

4. Systemverhalten und Benutzungsschnittstelle

Dieser Bereich beschreibt, wie sich das System dem Benutzer darstellen sollte, welche Funktionalität ihm zu bestimmten Zeitpunkten zur Verfügung stehen muß und wie es auf seine Eingaben reagieren soll. Er umfaßt also die Funktionen, bei denen das System aktiv agiert.

verschiedene Sichten

Der Benutzer muß die Möglichkeit haben sich das aktuelle Dokument in verschiedenen Sichten anzeigen zu lassen. Eine Sicht bezeichnet ein Bildschirmfenster, das das jeweilige Dokument in einer vorher eindeutig definierten Weise darstellt.

Zu diesen Sichten gehören mindestens:

- WYSIWYG-Sicht
- Steuerzeichensicht
- ASCII-Sicht
- PostScript-Sicht
- Struktursicht

Beschreibung der einzelnen Sichten

WYSIWYG

Der Übersetzung von „WYSIWYG“ (What You See Is What You Get) folgend sollte das Dokument hier nahezu so dargestellt werden, wie es nach dem Ausdrucken auf Papier tatsächlich aussieht. Dies gilt insbesondere für die Darstellung der im Kapitel „Dokumenten-Formatierung“ beschriebenen Mindestformatierungsmöglichkeiten.

Befindet sich der Benutzer im WYSIWYG-Fenster, d. h. der Text-Cursor wurde mit Hilfe des „grafischen Cursors“ im Fenster gesetzt, dann muß es ihm möglich sein:

- zu editieren
- alle in Kapitel „Dokumenten-Eingabe“ beschriebenen Funktionen auszuführen
- alle in Kapitel „Ein-/Ausgabeverhalten“ beschriebenen Funktionen auszuführen
- zu erkennen, wo im Text er sich befindet
- alle Sichten zu öffnen
- diese Sicht zu schließen

Steuerzeichensicht

Diese Sicht stellt das Dokument in der im Kapitel „Logische Struktur“ definierten „FORAUS“-Notation dar. Diese Notation ist SGML-ähnlich, d. h. der Text wird gemäß seiner logischen Struktur ausgezeichnet.

Die in der WYSIWYG-Sicht offensichtliche Formatierung verbirgt sich in der Steuerzeichensicht hinter den logischen Steuerzeichen.

Es muß dem Benutzer in der Steuerzeichensicht dieselbe Funktionalität wie in der WYSIWYG-Sicht zur Verfügung stehen. Zusätzlich sollte es möglich sein, mit Hilfe der Steuerzeichen ein logisches Element zu markieren. Der Benutzer darf ein logisches Steuerzeichen selbst nicht editieren.

Indexeinträge

Außerdem muß die Möglichkeit bestehen, innerhalb des Dokumentes Indexeinträge einzufügen. Aus den Indexeinträgen kann später ein Register oder ein Inhaltsverzeichnis erstellt werden.

Objekte einfügen (wünschenswert)

Wünschenswert wäre die Möglichkeit, Grafiken, Tabellen und ASCII-Texte im Dokument plazieren zu können. Es wäre von Vorteil, wenn die im Text eingefügten Grafiken ausgeschnitten, vergrößert, verkleinert und kopiert werden könnten. Die Grafiken sollten sowohl an eine feste Stelle im Dokument als auch an bestimmte Textpassagen gebunden werden können.

Der Benutzer soll ebenfalls über eine Möglichkeit verfügen, einen Textblock auszuschneiden. Ebenso wie beim Löschen muß vor dem Ausschneiden ein Textblock markiert werden. Der markierte Textblock wird anschließend aus dem Dokument entfernt und in einen Zwischenspeicher geschrieben.

Dem Benutzer muß eine Funktion zur Verfügung gestellt werden, um Textteile kopieren zu können. Auch beim Kopieren muß vorher ein Textblock markiert werden. Im Gegensatz zum Ausschneiden wird der markierte Textteil jedoch nicht aus dem Dokument entfernt, sondern lediglich in einen Zwischenspeicher geschrieben.

Der Inhalt des Zwischenspeichers muß an jeder beliebigen Stelle im Dokument eingefügt werden können. Umfaßt der markierte Textblock den gesamten Text eines oder mehrerer logischer Elemente, muß beim Einfügen, Ausschneiden und Löschen überprüft werden, ob durch die Aktion, die in der DTD beschriebene Syntax erhalten bleibt.

Einfügen

Das Einfügen eines neuen logischen Elementes muß ebenfalls möglich sein. Auch beim Einfügen soll überprüft werden, ob das neu eingefügte logische Element nach dem in der DTD beschriebenen syntaktischen Aufbau eingefügt werden darf. Notwendig ist auch, daß ein Datumsfeld, welches mit dem aktuellen Tagesdatum gefüllt wird, eingefügt werden kann.

UNDO und REDO

Um im Text durchgeführte Änderungen zurücknehmen zu können, muß der Benutzer über eine „UNDO“-Funktion verfügen. Wird die „UNDO“-Funktion aufgerufen, wird der Text in den Zustand versetzt, der vor der letzten Änderung gültig war. Eine Änderung umfaßt sämtliche Eingaben und Löschungen, die zwischen zwei Cursorbewegungen getätigt wurden. Durch wiederholte Betätigung des „UNDO“-Befehles muß der Benutzer solange Änderungen zurücknehmen können, bis er Zustand erreicht wurde, an dem das Dokument zum letzten Mal gespeichert wurde.

Passend zur „UNDO“-Funktion sollte der Benutzer auch über eine „REDO“-Funktion verfügen. Diese Funktion führt die zuletzt zurückgenommene Änderung erneut durch. Der „REDO“-Befehl darf jedoch nur angewendet werden, wenn nach dem Aufruf der „UNDO“-Funktion das Dokument nicht verändert wurde.

Abbrechen bei laufender Verarbeitung

Wird vom Benutzer eine Funktion aufgerufen, dessen Verarbeitung längere Zeit in Anspruch nimmt, sollte der Benutzer über eine Möglichkeit verfügen, die Verarbeitung abzubrechen. Wird eine Funktion abgebrochen, wird der Text in den Zustand versetzt, der vor dem Aufruf der abgebrochenen Funktion gültig war.

Querverweise

Der Benutzer muß darüber hinaus über Möglichkeiten verfügen, im Dokument Querverweise zu Textpassagen und logischen Elementen zu plazieren. Textpassagen und logische Elemente müssen deshalb mit einer Markierung versehen werden können, auf die später im Querverweis Bezug genommen wird. Der Benutzer sollte bestimmen können, ob im Querverweis die Seitenzahl, auf der sich die Textpassage bzw. das logische Element befindet, aufgeführt wird. Querverweise sollten sich nicht nur auf bestimmte logische Elemente beziehen können, sondern auch auf das letzte bzw. das nächste im Text vorkommende logische Element eines Elementtyps.

3. Dokumenten-Eingabe

Dieser Abschnitt beschreibt die Funktionen, bei denen der Benutzer aktiv mit dem System umgeht. Er enthält u. a. die Möglichkeiten der Eingabe und Änderungen des Dokumentes.

Bewegen des Cursors im Dokument

Um den Textcursor an eine bestimmte Stelle im Dokument zu bewegen, sollte der Benutzer über folgende Steuerungsmöglichkeiten verfügen.

Der Cursor muß sowohl mit der Maus, als auch mit der Tastatur bewegt werden können. Wurde der Cursor an eine Stelle bewegt, die über den rechten Rand einer Zeile hinausgeht, wird der Cursor automatisch auf das Ende dieser Zeile plaziert. Über spezielle Tastenkombinationen sollte der Benutzer den Cursor ein Zeichen, ein Wort oder eine Seite im Dokument vor- bzw. zurückbewegen können. Eine Möglichkeit, den Cursor an den Anfang des nächsten logischen Elementes zu plazieren, sollte ebenfalls vorgesehen sein. In einer Sicht, in der nur bestimmte Elemente angezeigt werden, ist so eine schnelle Bewegung durch das Dokument möglich. Den Cursor an den Anfang bzw. das Ende des Dokumentes zu setzen, muß möglich sein. Außerdem muß der Cursor an den Zeilenanfang oder das Zeilenende plaziert werden können.

Veränderung des Dokumentes

Bei der Eingabe eines Textes wird zwischen dem „Einfügen“- und dem „Überschreiben“-Modus unterschieden. Befindet sich das Programm im „Einfügen“-Modus, wird der neu eingegebene Text an der aktuellen Cursorposition eingefügt. Im „Überschreiben“-Modus wird das Zeichen, auf dem sich der Cursor befindet, durch das neu eingetippte Zeichen ersetzt. Befindet sich der Cursor am Absatzende, werden die eingegebenen Zeichen auch im „Überschreiben“-Modus eingefügt.

Suchen und Ersetzen

Dem Benutzer muß eine Möglichkeit zur Verfügung gestellt werden, Zeichenketten im Dokument zu suchen. Die gesuchte Zeichenkette sollte durch eine vom Benutzer einzugebende Zeichenkette ersetzt werden können. Dabei wäre es wünschenswert, wenn der Benutzer über eine Wahlmöglichkeit verfügen würde, ob vor jedem Ersetzungsvorgang eine Abfrage erfolgen soll.

Um einen Text einfacher ergänzen und verändern zu können, müssen dem Benutzer weitergehende Funktionen zur Textbearbeitung zur Verfügung gestellt werden.

Textteile markieren

Es muß möglich sein, den zu bearbeitenden Textteil zu markieren. Das Markieren sollte sowohl mit der Maus als auch mit der Tastatur möglich sein. Der Benutzer soll nicht nur einen zusammenhängenden Textteil, sondern auch ein oder mehrere logische Elemente markieren können. Wurde ein logisches Element markiert, umfaßt der Markierungsbereich den gesamten Text, der diesem logischen Element zugeordnet ist.

Operationen auf markierte Textteile

Das Löschen eines Textblocks muß möglich sein. Voraussetzung für das Löschen ist, daß vorher ein Textblock markiert wurde. Der markierte Textblock wird anschließend aus dem Dokument entfernt.

2. Logische Struktur

Ein Dokument besteht nicht nur aus den Zeichen und Objekten, die in ihm enthalten sind. Ein ebenfalls wichtiges Merkmal ist die ihm zugrundeliegende logische Struktur. Wie im Abschnitt „Dokumenten-Eingabe“ beschrieben, sind nur solche Eingaben möglich, die dieser logischen Struktur entsprechen. Jedem Dokument kann zur Zeit nur eine derartige Struktur zugrunde liegen.

Mit den logischen Elementen werden dann durch sog. Präsentationsregeln Formatierungsanweisungen verknüpft (siehe auch „Dokumenten-Formatierung“). Die Beschreibung der Präsentationsregeln hat geeignet zu erfolgen.

Zur Beschreibung der logischen Struktur dient eine Untermenge des SGML, die in einer sog. DTD erfolgt. Folgende Einschränkungen des SGML-Sprachumfanges werden zugelassen:

- die DTD soll getrennt vom Dokument in einem separaten File abgelegt werden.
- die hierarchische Struktur des Dokumentes wird durch Repräsentation der Textelemente und ihrer Abhängigkeiten entsprechend der DTD beschrieben.
- als Datentypen für die Textelemente der DTD sind zugelassen:
  - EMPTY (leeres Element)
  - #NDATA (Non-SGML-Data, z. B. Grafiken)
  - #PCDATA (beliebig lange Zeichenketten, inkl. der Länge Null)
- die Attributierung von Elementen ist zugelassen, hierfür gilt:
  - es gibt nur ein Referenzattribut mit einer eindeutigen ID, die die Referenz identifiziert
  - es gibt nur ein Attribut pro Element, das auf eine eindeutige REFID verweist
- Entitäten werden für den Fall unterstützt, daß auch Variablen vom Programm unterstützt werden, einschränkend gilt:
  - es werden keine Entitäten in Entitäten unterstützt
  - es werden keine Entitäten zur Umdefinierung von Zeichen aller Art und ihrer spezifischen Bedeutung unterstützt
  - Entitäten dürfen keine Markup-Sequenz enthalten
  - Entitäten dürfen keine Verarbeitungsanweisungen enthalten, die das System betreffen (z. B. Formatierungsanweisungen)
  - es werden nur einfache Entitäten unterstützt, die durch eine Textersetzung aufgelöst werden können und keine kompletten Markup-Anweisungen enthalten
- die Möglichkeit zur kontextabhängigen Sinndefinition von Zeichen und Zeichenketten wird nicht unterstützt.
- jegliche Art von Minimierung in Markups und DTD wird nicht unterstützt.

1. Einleitung

Eine Anforderungsdefinition stellt die Grundlage für die Systemspezifikation dar. Bevor genau beschrieben werden kann wie etwas gestaltet werden kann, muß vorher feststehen, was überhaupt gestaltet werden soll. Das „was“ etwas genauer zu erläutern ist das Ziel dieses Dokumentes.

Es entstand auf der Grundlage der Ausarbeitung „Anforderung an ein innovatives Dokumentenverarbeitungssystem“ und entsprechenden Absprachen im Plenum. Aus diesem Grund kann an dieser Stelle noch keine Vollständigkeit in allen Bereichen vorausgesetzt werden, da die Grundlagen selber diesem Anspruch nicht gerecht werden. Vielmehr wird es so sein, daß im Laufe der Spezifizierung hier noch nicht erfaßte Bereiche und Problemfälle auftreten werden, die dann an diesen Stellen mit den jeweils betroffenen Teilbereichen abgestimmt werden müssen.

Die Anforderungsdefinition ist in 5 Bereiche unterteilt. Zuerst werden Anforderungen an die „logische Struktur“ erläutert. Hier wird die Grundlage für das gesamte Dokumentenverarbeitungssystem gelegt. Die wichtigsten Stichwörter sind hier SGML, DTD und Präsentationsregeln.

Im zweiten Abschnitt „Dokumenten-Eingabe“ werden die Anforderungen genannt, die sich aus Funktionen ergeben, bei denen der Benutzer selber aktiv ist. Hierunter fallen u. a. alle Eingaben und Textmanipulationen.

Weiter geht es mit den Funktionen, bei denen vorwiegend das System „aktiv“ ist. Der Benutzer selber kann diese nicht direkt durchführen. Dieser Abschnitt befaßt sich also mit „Systemverhalten und Benutzungsschnittstelle“.

„Dokument-Formatierung“ stellt nun die Anforderungen dar, die an die Formatierung gestellt wird. Diese Funktionen stellen zwar im engeren Sinne ebenfalls ein Sytemverhalten dar, sie wurden jedoch in einem eigenen Kapitel zusammengefaßt.

Als letzten Bereich werden kurz die Anforderungen an das „Ein-/Ausgabeverhalten“ aufgezählt. Diese beziehen sich auf die Datei- und Druck-Funktionen.

In den Abschnitten sind am Ende die Funktionalitäten aufgeführt, die im Plenum mit einer gegenüber den anderen Funktionen niedrigeren Priorität versehen wurden. Ihnen wurde jeweils „(wünschenswert)“ angehängt.