Bericht zur Semesterarbeit
MEDINO

Nikolaos Kaintantzis

11. September 2000

Inhaltsverzeichnis

1 Zusammenfassung **1**

2 Was ist MEDINO? **4**

3 Aufgabenstellung **6**
3.1 Idee 6
3.2 Aufgabenstellung für die Semesterarbeit 6

4 Konzeption **7**
4.1 CD statt Webseite 7
4.2 Benutzerführung 8
4.2.1 Willkommensseite 8
4.2.2 Hauptseite 9
4.2.3 Architekturplan eines Spitals 10
4.2.4 Arbeitsablauf im OP eines Spitals 11
4.2.5 Wissensthemen 13
4.3 Entwicklungsgeschichte 16
4.3.1 Varianten für die Gesamtnavigation 16
4.3.2 Varianten für die Wissensdaten 17
4.4 Page-Generator für die Wissensdaten 18
4.5 gelöste infomatische Probleme 20
4.6 Ausblick 21

5 Ergonomie-Prinzipien für Software **22**
5.1 Aufgabenangemessenheit 23
5.2 Selbstbeschreibungsfähigkeit 23
5.3 Steuerbarkeit 23
5.4 Erwartungskonformität 24
5.5 Fehlertoleranz 24
5.6 Individualisierbarkeit 25
5.7 Lerneignung 25

6 unterstützte Browser **26**
6.1 theoretisch unterstützte Browser 26
6.2 getestete Browser 26
6.3 mögliche Probleme im Betrieb 27

7 Erweiterung von MEDINO (Anleitung) **29**
7.1 Organisation der Files 29
7.2 Spitalauswahl (Wahl des Arbeitssystems) 30
7.3 Architektur eines Spitals 31
7.4 Arbeitsablauf in einem Spital 32
7.5 Wissensthemen 33

A Glossar **36**

B Quellcodes **38**
B.1 TextBox.java 38
B.2 ablauf.java 39
B.3 intro.java 41
B.4 welcome.java 43
B.5 makepage.java 44

C Pages **55**
C.1 Willkommensseite 55
C.2 Hauptseite 57
C.3 Architektur 65
C.4 Arbeitsabläufe 67
C.5 Wissensthemen 68
C.5.1 Vergleichsmodus 68
C.5.2 Gesammt-Navigation 70
C.5.3 Spital und Berufsgruppenwahl 72

Abbildungsverzeichnis

4.1 Gesamtnavigation in MEDINO 8
4.2 Willkommensseite 8
4.3 Willkommensseite: Java und JavaScript sind ausgeschaltet . . 9
4.4 Hauptseite 10
4.5 Architekturplan des Inselspitals 11
4.6 Arbeitsablauf im Inselspitals 12
4.7 Legende zum Arbeitsablauf 12
4.8 Wissensthemen 14
4.9 Vergleichsmodus Wissensthemen 15
4.10 Feedbackfomular zu den Wissensthemen 15
4.11 Erste Version der Gesamtnavigation 16
4.12 Zweite Version der Gesamtnavigation 16
4.13 Variante mehr Abstand 17
4.14 Punktsymmetrische Anordnung 18
4.15 Vorlagen-File im Word 18

7.1 Organisation der Files in MEDINO 29
7.2 Erfolgreiche Ausführung von makepage 33
7.3 Laden der Vorlagedatei in makepage 34
7.4 Zieldatei sichern im Arbeitsordner von makepage 35

Kapitel 1

Zusammenfassung

MEDINO ist eine Visualisierung von Betriebsabläufen und eine Präsentation von Benutzerwissen über Arbeitssysteme. In dieser Semesterarbeit war der Fokus auf OP-Trakte von vier Spitälern gerichtet.

Die Zielgruppen sind Architekten, die sich mit Spitälern befassen und somit die Arbeitsumgebung gestalten als auch Arbeitspersonen in den OP-Trakten, die sich für die Arbeitsorganisation in Spitälern interessieren.

Die Datenerfassung für MEDINO geschah mittels objektgestützten Interviews. In diesen besonderen Interview-Situationen kommen Hilfen zur Explikation von Erfahrungswissen zum Einsatz (VALAMO-Methode). Dadurch kommen auch Probleme zum Vorschein, welche die Personen in einem konventionellen Interview nicht erwähnt hätten. Auf diese Methode der Datenerhebung wird in der Studie "OP-Organisationskonzepte im Vergleich – Eine Informations- und Betriebplanungsgrundlage" von Dr. Jürgen Held genauer eingegangen.

Die Aufgabenstellung dieser Semesterarbeit war die Umsetzung der Daten als ergonomisch zu bedienendes Informationssystem.

Beim Start von MEDINO erscheint die "Willkommens-Seite". Diese, auf den ersten Blick schlichte Seite, ist der Schlüssel für einen reibungsfreien Ablauf innerhalb von MEDINO. Die Willkommens-Seite arbeitet wie ein Türsteher. Genügt der Browser den Anforderungen von MEDINO, wird der Eingang zu MEDINO gezeigt (Link zur Hauptseite wird eingeblendet) und man darf eintreten. Genügt er den Anforderungen nicht, wird eine Erklärung geliefert, wieso man nicht rein darf, und der Eingang wird nicht gezeigt (kein Link zur Hauptseite).
Durch die Willkommensseite wird eine Vorselektion der Browser getroffen. Wer reinkommt, wird später auf keine Probleme im Innern treffen.

Die Navigation zwischen den Seiten ist stark vereinfacht. Der/Die BenutzerIn weiss zu jeder Zeit, wo er/sie sich befindet, da jede Seite eine Überschrift besitzt.

Von der Haupseite aus (siehe auch Abbildung 4.1 auf Seite 8 im Kapitel Benutzerführung) verzweigt man in den gewünschten Bereich. Zur Auswahl stehen die Arbeitssysteme und die Wissensthemen. Von jeder Seite aus kann man via Textlink zur Hauptseite zurück.

Der Bereich Arbeitssysteme umfasst Architektur und Arbeitsabläufe. Zum gewünschten Arbeitssystem (Spital) gelangt man über die Selektion der Luftaufnahme des entsprechenden Spitals.
Im Teilbereich Architektur erhält man Einsicht in die ursprüngliche Nutzungsplanung als auch in die tatsächliche Verwendung der Räume. Die Pläne sind per Maus begehbar. Es werden Erklärungen eingeblendet, wenn man interessante Räume oder Gänge betritt.
Im Teilbereich Arbeitsabläufe, werden die Arbeitsbedingungen visualisiert. Dies geschieht mittels einer Diashow.

Die Wissensthemen werden auf der Hauptseite durch eine Mindmap vorgestellt. Der/Die BenutzerIn kann sich bereits hier mit den Themen vertraut machen. Befindet sich der Mauszeiger über einem Ast, so wird unterhalb der Mindmap eine Beschreibung eingeblendet. Klickt man auf den Ast gelangt man in den Bereich Wissensthemen, in welchem die Daten zu diesem "Ast" (Wissenspunkt) anzeigt werden.

Um den Vergleich zwischen den Spitälern im Bereich Wissensthemen zu vereinfachen, wurde der Vergleichsmodus entwickelt. Der/Die BenutzerIn kann durch einen Mausklick in diesen schalten und dort z.B. zwischen Berufsgruppen und Spitäler vergleichen.
Bei den Arbeitssystemen muss der/die BenutzerIn selbst vor und zurück navigieren, um Vergleiche anzustellen. Dies ist ihm/ihr aber zuzumuten, da es sich bei den Anwendern und Anwenderinnen um Fachpersonen handelt, die sich mit den typischen Problemen in den Arbeitssystemen auskennen und sich diverse Details merken können.

MEDINO wurde aus informatischer Sicht so gestaltet, dass die Person, welche eine allfällige Erweiterung durchführen will (z.B. neues Spital einfügen), lediglich HTML-Kenntnisse besitzen muss, um den Code anzupassen.

Beim Testen der CD sind die Schwächen in den Filesystemen der einzelnen Betriebssystemen zum Vorschein gekommen. Dies hat dazu geführt, dass wir MEDINO als Windows-CD anbieten. Zusätzlich befindet sich ein Zip-Archiv auf der CD. Um MEDINO auf andere Betriebssysteme zu benutzen, kann man dieses Zip-Archiv entpacken oder die MEDINO-Webseiten besuchen.

MacIntosh beispielsweise unterstützt 256-Zeichen-lange Dateinamen, kann aber in der Praxis lediglich mit maximal 32-Zeichen-langen Dateinamen umgehen. Dadurch kann die MEDINO-CD auf dem Macintosh nicht benutzt werden. Man kann also nur warten, bis der Macintosh 256-Zeichen-lange Dateinamen wirklich unterstützt.

Bei der Programmierung wurde darauf geachtet, dass auch ältere Browser benutzt werden können. Dennoch wurde bei Tests festgestellt, dass die Browser auf dem Macintosh, trotzt gleicher Versionsnummern wie diejenigen für Windows, nicht die volle Funktionalität bieten. MEDINO kann auf Macs erst benutzt werden, wenn neue (fehlerfreie) Browserversionen auf den Markt kommen.

Mit der Semesterarbeit MEDINO ist nicht nur dieses Informationssystem oder nur diese CD entstanden. Die besondere Strukturierung und der benutzerfreundliche Aufbau von MEDINO lässt sich als Gerüst für weitere MEDINOs mit anderen Arbeitssystemen (zum Beispiel aus der Industrie) verwenden.

Kapitel 2

Was ist MEDINO?

MEDINO steht für Management, Ergonomie, Design und Information für Organisationen.

Das Besondere an MEDINO ist nebst der ergonomischen Gestaltung die Datenerfassung. Daten wurden mittels objektgestützten Interviews erhoben (VALAMO-Methode).

Beim Interview stand eine Metalltafel/Metalltablett zur Verfügung, auf welchem die benötigten Grundrisse massstabsgetreu befestigt waren. Auf diesem Tablett konnte man nun Magnetsteine in unterschiedlichen Farben und Formen verschieben.

Die VALAMO-Methode ermöglicht ein besseres Verbalisieren des Wissens. Durch das Zeigen und bewegen der Objekte wird das Wissen leichter transportierbar und übermittelbar. Der ausserhalb des Arbeitssystems stehende Interviewer hat so direkten Einblick in die Handlungen. Dadurch zeigen sich auch Probleme, die der befragten Person gar nicht mehr bewusst waren, da sie sich schon an deren Vorhandensein gewöhnt haben.

Ein weiterer Vorteil ist das Diskutieren von Szenarien. "Was wäre wenn ..."-Fragen, die manchmal das Vorstellungsvermögen strapazieren, kann man hier durchgehen. Die Situation hat man Dank des Tabletts mit den Magnetsteinen immer vor Augen und in Griffnähe.

Diese Daten wurden in der Studie "OP-Organisationskonzepte im Vergleich – Eine Informations- und Betriebplanungsgrundlage" von Dr. Jürgen Held erhoben, die er im Auftrag des Inselspital Bern durchgeführt hat.

MEDINO baut auf dieser Studie auf und ist als Semesterarbeit im Nebenfach "Arbeitswissenschaften" am Institut für Hygiene und Arbeitspsychologie, innerhalb des Fachbereichs Ergonomie, Arbeit und Gesundheit entstanden.

Dieser Bericht dokumentiert die Entstehung von MEDINO, bewertet das Resultat und erklärt, wie man MEDINO selbst erweitern kann.

Kapitel 3

Aufgabenstellung

3.1 Idee

Frage: Wie können weltweit Erfahrungen über Arbeitsabläufe und -gestaltung ausgetauscht werden?

Antwort/Aufgabe: Mittels einer WWW-Page zur Anzeige und Ergänzung von Daten! Es liegen schon strukturierte und elektronisch gespeicherte Daten aus Interviews über Arbeitsabläufe von OP-Abteilungen der Mayo-Klinik (USA), der Uniklinik Leuven (Belgien), dem CHUV (Lausanne), dem Inselspital (Bern) und dem Unispital (Zürich) vor. Diese Daten sollen mit einer ergonomisch(!) gestalteten WWW-Page abzufragen und zu ergänzen sein.

Die ursprüngliche Idee wurde in Projektsitzungen konkretisiert. Dabei wurden mehrere Varianten diskutiert. Die eigentliche Aufgabenstellung wurde während der Semesterarbeit spezifiziert.

3.2 Aufgabenstellung für die Semesterarbeit

- Präsentation der Architekturpläne. Der/Die BenutzerIn kann mit der Maus über die jeweiligen Grundrisse fahren. Befindet sich die Maus über einem sensitiven Bereich, erscheint der zum Bildbereich gehörende Text.
- grafische Präsentation der Arbeitsabläufe in den jeweiligen OP-Sälen.
- Interview-Daten auf einer Page strukturiert darstellen. Die Aussagen sollen spitalweise verglichen werden können.

Kapitel 4

Konzeption

4.1 CD statt Webseite

Durch die Wahl, eine CD statt eine Webseite zu erstellen, musste auf die automatische Erweiterbarkeit der Wissensdaten verzichtet werden. MEDINO wird nun erweitert, indem es neue Versionen gibt. Um an neue Daten zu kommen, wurde ein Feedbackformular geschrieben. Der User kann aber seine Erfahrungen auch via Email zusenden.

Des weiteren musste aber auch eine Lösung gefunden werden, die Daten zu managen. Eine Datenbank war nicht mehr möglich, da alles auf dem lokalen Computer des Anwenders / der Anwenderin laufen muss.

Als Hilfsmittel zu HTML konnten nur Java, JavaScript und Cookies verwendet werden. Serverseitige Sprachen wie php oder jsp fielen weg. Damit MEDINO auf möglichst vielen Plattformen und Browsern läuft, musste darauf geachtet werden, auf möglichst tiefe Versionen der verwendeten Hilfsmittel zuzugreifen. Natürlich musste auf browserspezifische "Befehle" verzichtet werden, sofern sie nur von einem Browser unterstützt werden.

Die CD hat den grossen Vorteil, dass die Ladezeit nicht ausschlaggebend ist. Somit konnte die Qualität der Bilder hoch gewählt werden, was für die Arbeitsabläufe sehr wichtig ist. Ebenfalls konnte man vermehrt mit grafischen Effekten arbeiten.

4.2 Benutzerführung

Die Abbildung 4.1 zeigt schematisch die aktuelle Gesamtnavigation innerhalb von MEDINO. Der/Die BenutzerIn ladet die Willkommensseite.

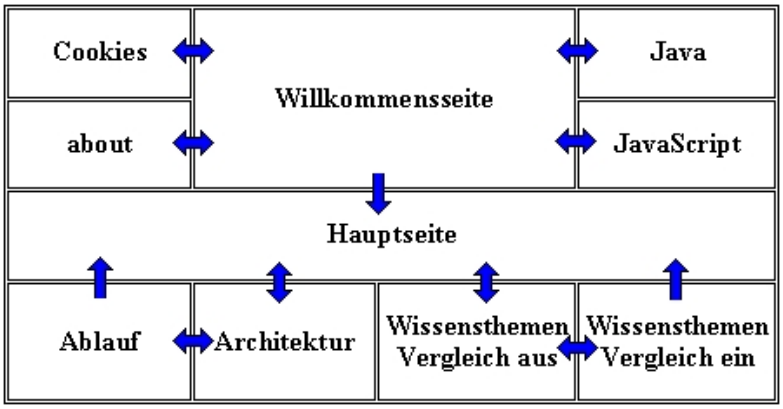


Abbildung 4.1: Gesamtnavigation in MEDINO

4.2.1 Willkommensseite

Abbildung 4.2: Willkommensseite

Abbildung 4.3: Willkommensseite: Java und JavaScript sind ausgeschaltet

Die Willkommensseite (Abbildung 4.2) überprüft beim Laden, ob alle verwendeten Hilfsmittel[1] zur Verfügung stehen. Wenn dem nicht so ist, erfolgt die Bitte, diese einzuschalten (siehe Abbildung 4.3), gefolgt von einem Link zur Informationsseite des jeweiligen Hilfsmittels. Auf einer Informationsseite wird erklärt, für welche Zwecke das entsprechende Hilfsmittel eingesetzt wird.

Über einen Link (oder durch einen Klick aufs Bild) gelangt man zur Hauptseite. Ein weiterer Link führt in den About-Bereich, in welchem Hintergrundinfos über MEDINO zu lesen sind.

4.2.2 Hauptseite

Die Hauptseite (Abbildung 4.4) ist zweigeteilt. In der linken Hälfte befinden sich Luftaufnahmen von Spitälern. Bewegt man den Mauszeiger über ein Bild, verändert es sich (wird ausgefüllter). Es entsteht der Eindruck, man hätte eine Auswahl getroffen. Klickt man nun auf das hervorgehobene Bild, gelangt man zur entsprechenden Architektur-Seite des Spitals.
In der rechten Hälfte ist die Mindmap untergebracht. Befindet sich die Maus über einem Ast, so wird unterhalb der Mindmap die Bedeutung dieses Punktes (eine Erklärung) eingeblendet. Klickt man nun auf diesen Ast, gelangt man zu den Wissensthemen. Es ist der entsprechende Wissenspunkt geöffnet (und zwar jener des zuletzt betrachteten Spitals und der zuletzt betrachteten Berufsgruppe). Der Vergleichsmodus ist ausgeschaltet.

[1] Java, JavaScript und Cookies

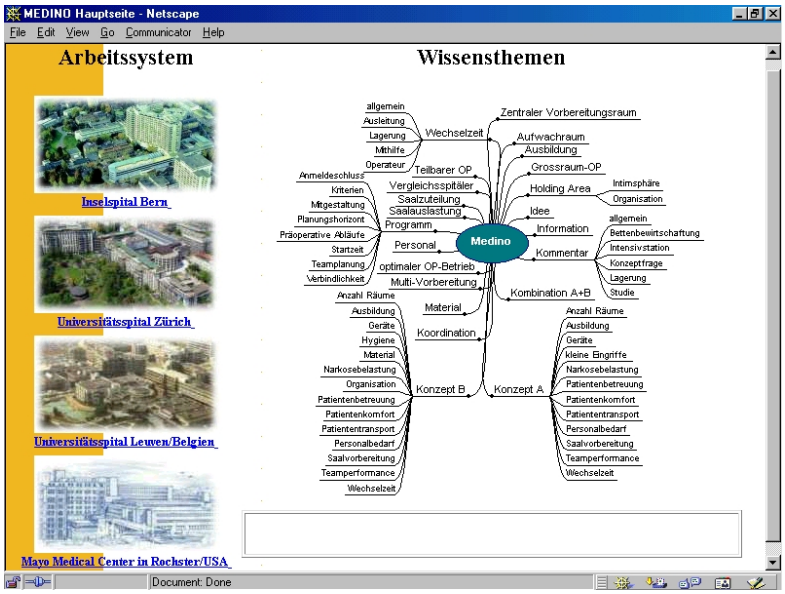


Abbildung 4.4: Hauptseite

Klickt man in die Mitte der Mindmap, wo sich der Schriftzug MEDINO auf grünem Hintergrund befindet, so gelangt man zum About-Teil.

4.2.3 Architekturplan eines Spitals

Die Entwicklung dieser Seite zeigte sich aufwendiger als gedacht. Zur Erinnerung: Es soll der Grundriss des OP-Trakts[2] angezeigt werden. Der/Die BenutzerIn kann mit der Maus über das Bild fahren und erhält Beschreibungstexte zum Bildbereich in welchem sich die Maus befindet.

Auch wenn es sich einfach anhört, das Problem liegt darin, dass es keine standardisierten Javascript-Objekte gibt, welche den gewünschten Effekte durchführen konnten.[3] Einzig der Internet-Explorer kannte das benötigte Objekt "all".

[2]Nebst dem Grundriss des OP-Trakts wird im MEDINO der Grundriss eines OP-Saals angezeigt.

[3]Die Möglichkeit die Texte als Bilder abzuspeichern, wurde ausgeschlossen. Diese hat zwar den Vorteil, dass sie leicht umzusetzen gewesen wäre, aber den Nachteil, dass ein nachträgliches Ändern des Textes oder Erweiterungen von MEDINO um weiter Spitäler mit hohem Aufwand verbunden gewesen wäre. Werden Texte als Bilder abgespeichert, sind sie für alle BenutzerInnen gleich, also auch die Schriftgrösse. Dies ist ein weiterer Nachteil.

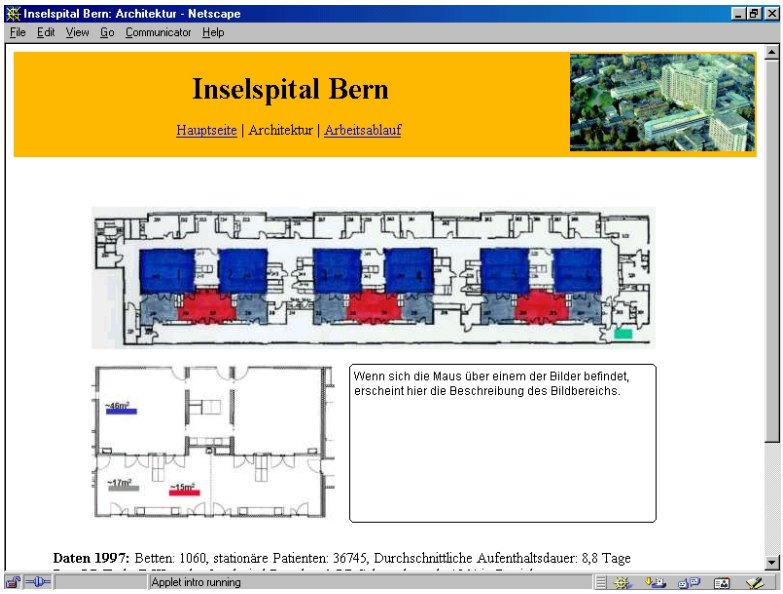


Abbildung 4.5: Architekturplan des Inselspitals

Das Problem wurde nun gelöst, indem Java verwendet wurde. Ein Applet musste geschrieben werden, das die zwei Grundrisse anzeigt und einen Text ausgiebt, sobald der Mauszeiger über einer sensitiver Stelle ist.

Das Problem mit Applets ist aber, dass es gewisse Restriktionen gibt, was sie auf einem fremden Computer machen dürfen und was nicht. Dies hat Folgen auf den Ablageort der Bilddaten, damit sie Java auch lesen und anzeigen darf. Mehr dazu im Kapitel 7 auf Seite 29 als auch im Kapitel 6.3 auf Seite 27.

Unterhalb der Überschrift befinden sich Links zur Navigation. Nebst dem obligaten Link zurück zur Hauptseite befindet sich ein Link zu den Arbeitsabläufen.

4.2.4 Arbeitsablauf im OP eines Spitals

Auch hier wurde ein Applet geschrieben. Nur so lässt sich eine Animation oder eine Dia-Show realisieren.

Als erstes wurde ein Applet entworfen, das eine Art Dia-Show durchführt (siehe Abbildung 4.6). Man sieht ein Bild des Arbeitsablaufes und den dazugehörenden Text. Zu jeder Zeit kann man ein Bild weiter (beim letzten Bild kommt wieder das erste), ein Bild zurück (beim ersten bliebt das erste) und

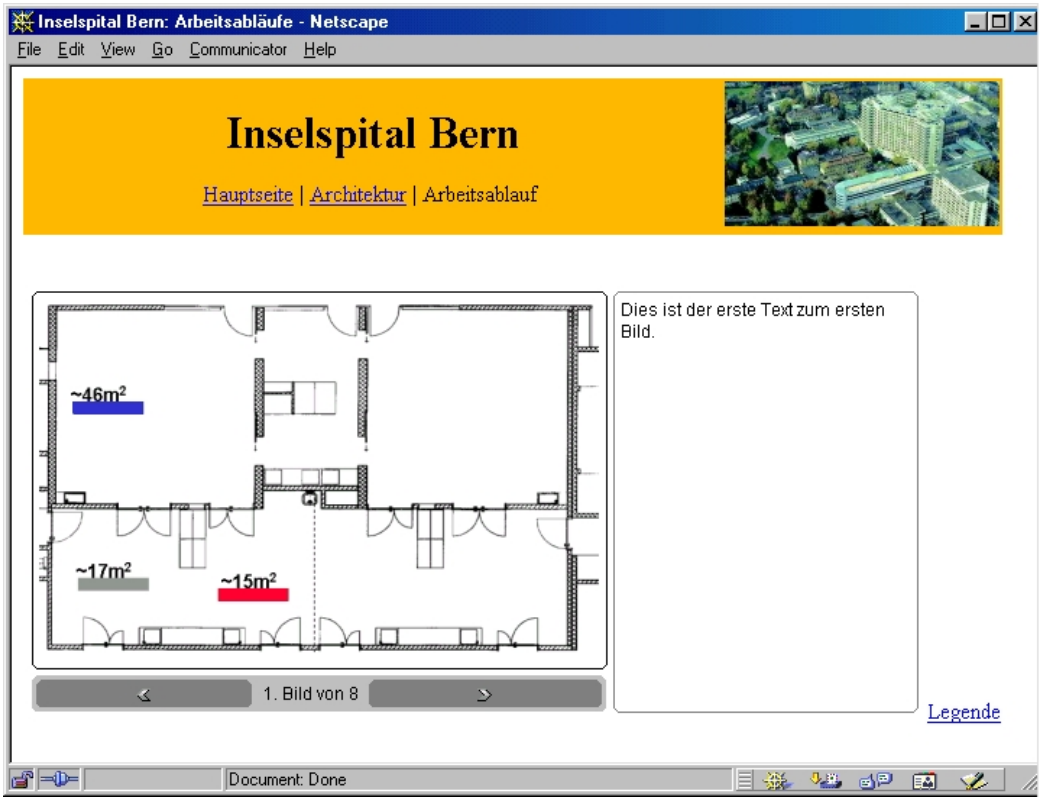


Abbildung 4.6: Arbeitsablauf im Inselspitals

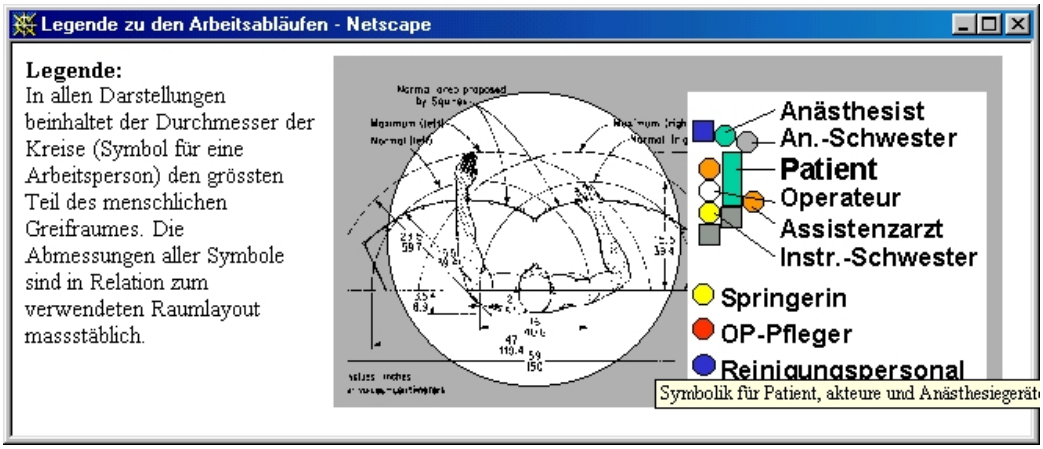


Abbildung 4.7: Legende zum Arbeitsablauf

sich eine Legende (siehe Abbildung 4.7) einblenden lassen. Mit dieser wird erklärt, dass die Kreise die Arbeitspersonen darstellen und die Rechtecke die OP-Tische mit den Patienten darstellen.
Beim Konzept der Dia-Show blieben wir auch.

Eine weitere Lösung wäre eine Art Daumenkino oder automatische Dia-Show. Diese Idee wurde verworfen, da hier das Tempo vorgegeben wäre. Um den Effekt bewegter Bilder zu erreichen, bräuchte es auch wesentlich mehr Bilder.

Die weitere Navigation ist ähnlich derjenigen bei der Architektur. Man hat einen Link zur Hauptseite und einen Link zur Architektur.

4.2.5 Wissensthemen

Dies war der umfangreichste und schwierigste Teil der Arbeit. Hier galt es die Datenmenge geschickt zu verwalten und ergonomisch zu präsentieren. Zugleich muss der User die Möglichkeit haben, die Daten zwischen den Spitälern vergleichen zu können.

Wir waren uns nicht sicher, ob die BenutzerInnen die Daten vergleichen wollen. Zugleich ist die Bedienung des Vergleichsmodus nicht für jeden/jede intuitiv, obwohl wir verschiedene Ansätze versucht haben (siehe Abschnitt 4.3.2 auf Seite 17). Aus diesem Grund haben wir uns entschlossen, den Vergleichsmodus als Option anzubieten.

Wie üblich befinden sich unterhalb der Überschrift Links zur Gesamtnavigation. Nebst dem obligaten Link zurück zur Hauptseite kann man in den Vergleichsmodus wechseln, eigene Kommentare via einem Feedback-Formular ans MEDINO-Team senden oder eine Hilfe zur Seite einblenden lassen.

Auf der linken Seite ist die Navigation innerhalb der Wissensthemen angebracht. Der/Die BenutzerIn kann das Spital, die Berufsgruppe und den betrachteten Wissenspunkt selbst bestimmen. Standardmässig ausgewählt ist der Wissenspunkt, den der/die BenutzerIn in der Mindmap bestimmt hat. Beim Spital und der Berufsgruppe ist ausgewählt, was der/die BenutzerIn beim letztmaligen Aufruf selektiert hatte. Gebraucht der User MEDINO zu ersten Mal oder hat er MEDINO seit mehr als einer Stunde nicht bedient, kommt die Standardeinstellung “Inselspital” und “alle Berufsgruppen”.

Wird das Spital oder die Berufsgruppe gewechselt und gibt es zur neuen Konstellation keine Aussage zum Wissenspunkt, so schrumpft die Wissensthemenauswahl auf diejenigen Themen, die wirklich angeboten werden. Somit

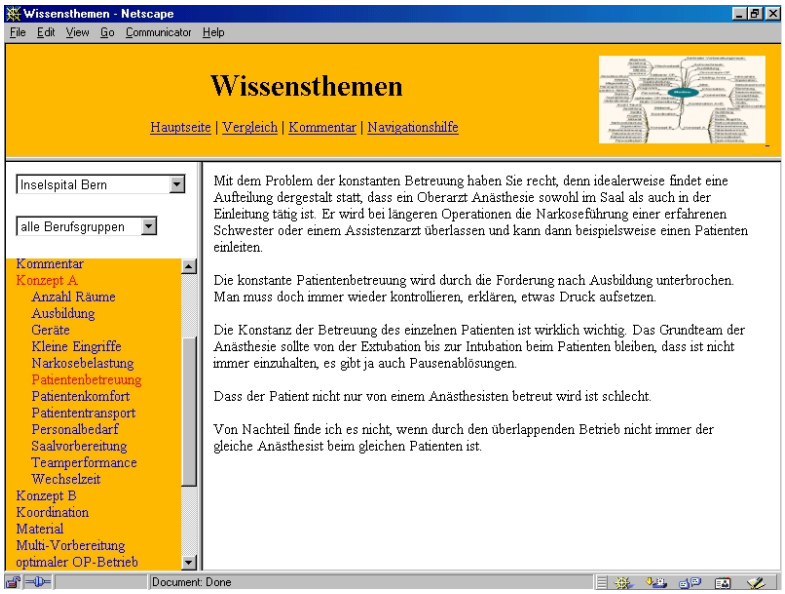


Abbildung 4.8: Wissensthemen

wird erreicht, dass der/die BenutzerIn nicht unnötig auf die Geduldsprobe gestellt wird.[4]

Der Vergleichmodus bietet zusätzlich zur Navigation auf der linken Seite eine identisch aussehende Navigation auf der rechten Seite an. Das Textfeld ist nun aufgeteilt in ein oberes und ein unteres. Das Benutzerwissen zur Auswahl auf der rechten Seite erscheint nun im unteren, das jenige auf der linken Seite im oberen Textfeld.

Im Feedbackformular ist automatisch derjenige Wissenspunkt ausgewählt, den der/die BenutzerIn zuvor betrachtet hatte. In einem weiteren Feld kann er/sie einen neuen Wissenspunkt vorschlagen, falls kein passender Punkt zu seiner Aussage existiert. Die weiteren Felder beziehen sich auf Angaben, die benötigt werden, um diesen Vorschlag in der nächsten Version von MEDINO einzubauen.
Die getätigten Angaben werden via Email versendet.

[4]Bei einigen Spitälern gibt es Berufsgruppen, die praktisch nicht befragt worden sind. Wären die Menüs nicht dynamisch und würde der/die BenutzerIn zufällig zuerst obige Konstellation auswählen und nach Aussagen suchen, würde er/sie mehrheitlich auf keine Aussagen stossen, was zu entnervtem Aufgeben und niemaligem Wiederbenutzen von MEDINO führen könnte.

Abbildung 4.9: Vergleichsmodus Wissensthemen

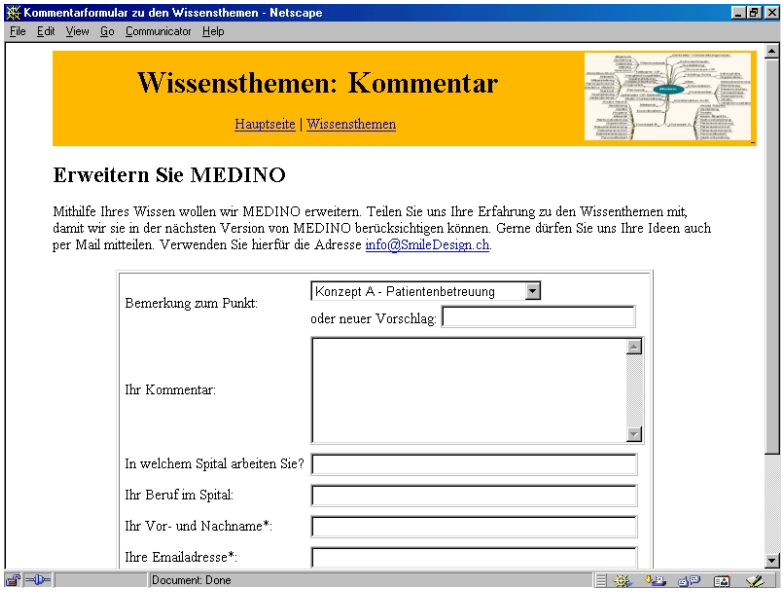


Abbildung 4.10: Feedbackfomular zu den Wissensthemen

4.3 Entwicklungsgeschichte

4.3.1 Varianten für die Gesamtnavigation

Das endgültige Layout der Gesamtnavigation ist nach langen Diskussionen entstanden. Dieser Abschnitt dokumentiert die Ideen.

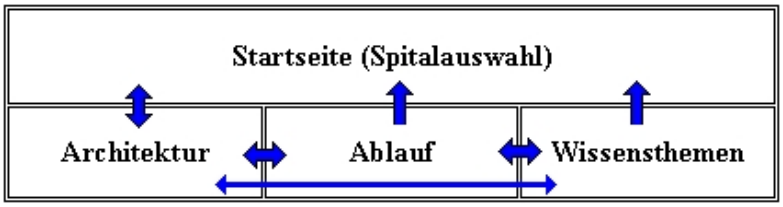


Abbildung 4.11: Erste Version der Gesamtnavigation

Die erste Version von MEDINO hatte die in Abbildung 4.11 aufgeführte Navigation. Man startete mit der Spitalauswahl. Entschied man sich für ein Spital gelangte man zur Architekturseite des Spitals. Von dort aus kam man via Links in die anderen Bereiche. Ein grosser Nachteil dieser Navigation war, dass der Vergleich der Wissensthemen zwischen den Spitälern mühsam war.

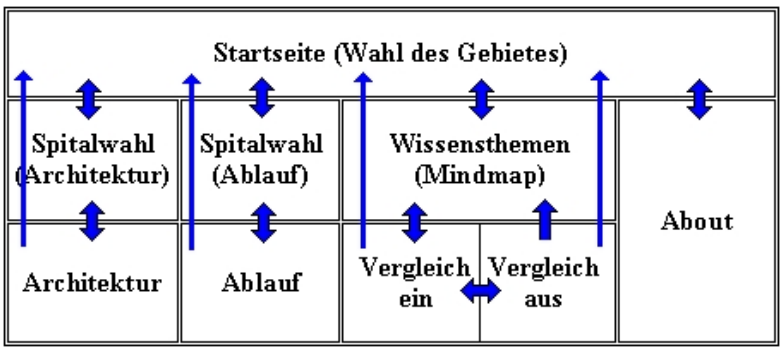


Abbildung 4.12: Zweite Version der Gesamtnavigation

Diesen Nachteil eliminierten wir mit der zweiten Version (Abbildung 4.12). Von der Startseite aus konnte man sich nun für die drei Gebiete, die MEDINO anbietet, oder die Hintergrundinformationen entscheiden.
Für die Wissensthemen gab es nun den Vergleichsmodus, der es einem erlaubte, die Aussagen zwischen den einzelnen Spitälern oder zwischen den Berufsgruppen zu vergleichen. Zusätzlich kam nun eine Mindmap hinzu.

Die obige Struktur hatte den Nachteil, dass die Navigation kompliziert wurde. Man hatte zu viele Möglichkeiten hin- und herzuspringen. Die Benennung der Links war auch nicht einfach. Zudem gab es die Spitalauswahl zweimal. In welcher Spitalauswahl man wirklich war, konnte man nur anhand der Überschrift eruieren.

4.3.2 Varianten für die Wissensdaten

Dieser Abschnitt befasst sich mit den verschieden Varianten des Vergleichsmodus der Wissensdaten. Das Hauptproblem war, wie man es dem User klar macht, welcher Text zu welchen Navigationselemente gehört.

Abbildung 4.13 zeigt eine Version, welche durch dickere Rahmen versuchte zu gruppieren. Dadurch sollte die Zugehörigkeit klarer werden. Ein Nachteil war allerdings, dass jetzt die Texte versetzt waren und das vergleichende Lesen mühsam war.

Abbildung 4.13: Variante mehr Abstand

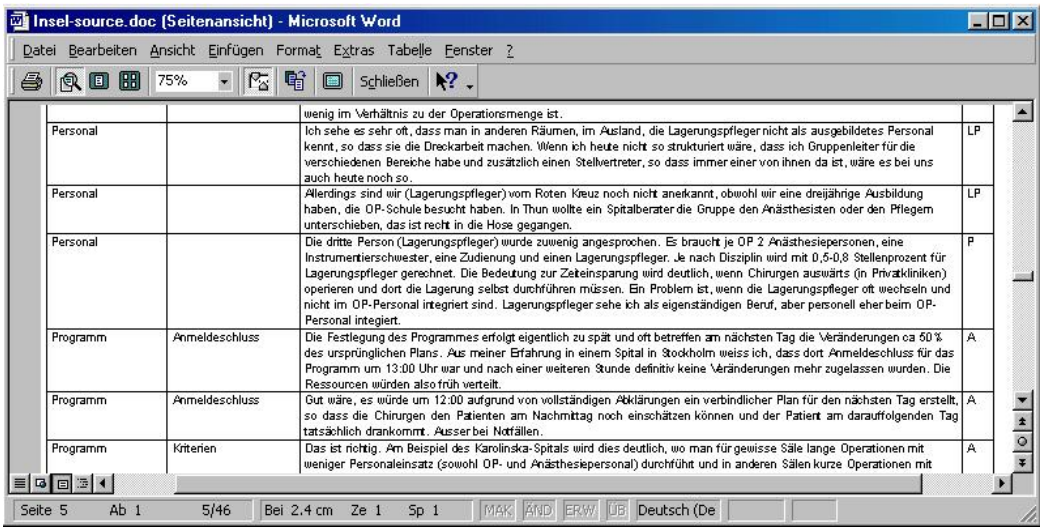
Eine weitere Idee ist in Abbildung 4.14 dargestellt. Alle zusätzlichen Elemente sind punktsymmetrisch angeordnet. Dadurch war die Zugehörigkeit klar. Doch nun war die Bedienung der rechten Navigation seltsam.

Des weiteren wurde auch hier mit dickeren Rahmen experimentiert.

Abbildung 4.14: Punktsymmetrische Anordnung

4.4 Page-Generator für die Wissensdaten

Die Rohdaten befinden sich pro Spital in einem eigenen Word-Dokument (siehe Abbildung 4.15).



		wenig im Verhältnis zu der Operationsmenge ist.	
Personal		Ich sehe es sehr oft, dass man in anderen Räumen, im Ausland, die Lagerungspfleger nicht als ausgebildetes Personal kennt, so dass sie die Dreckarbeit machen. Wenn ich heute nicht so strukturiert wäre, dass ich Gruppenleiter für die verschiedenen Bereiche habe und zusätzlich einen Stellvertreter, so dass immer einer von ihnen da ist, wäre es bei uns auch heute noch so.	LP
Personal		Allerdings sind wir (Lagerungspfleger) vom Roten Kreuz noch nicht anerkannt, obwohl wir eine dreijährige Ausbildung haben, die OP-Schule besucht haben. In Thun wollte ein Spitalberater die Gruppe den Anästhesisten oder den Pflegern unterschieben, das ist recht in die Hose gegangen.	LP
Personal		Die dritte Person (Lagerungspfleger) wurde zuwenig angesprochen. Es braucht je OP 2 Anästhesiepersonen, eine Instrumentierschwester, eine Zudienung und einen Lagerungspfleger. Je nach Disziplin wird mit 0,5-0,8 Stellenprozent für Lagerungspfleger gerechnet. Die Bedeutung zur Zeiteinsparung wird deutlich, wenn Chirurgen auswärts (in Privatkliniken) operieren und dort die Lagerung selbst durchführen müssen. Ein Problem ist, wenn die Lagerungspfleger oft wechseln und nicht im OP-Personal integriert sind. Lagerungspfleger sehe ich als eigenständigen Beruf, aber personell eher beim OP-Personal integriert.	P
Programm	Anmeldeschluss	Die Festlegung des Programmes erfolgt eigentlich zu spät und oft betreffen am nächsten Tag die Veränderungen ca 50 % des ursprünglichen Plans. Aus meiner Erfahrung in einem Spital in Stockholm weiss ich, dass dort Anmeldeschluss für das Programm um 13:00 Uhr war und nach einer weiteren Stunde definitiv keine Veränderungen mehr zugelassen wurden. Die Ressourcen würden also früh verteilt.	A
Programm	Anmeldeschluss	Gut wäre, es würde um 12:00 aufgrund von vollständigen Abklärungen ein verbindlicher Plan für den nächsten Tag erstellt, so dass die Chirurgen den Patienten am Nachmittag noch einschätzen können und der Patient am darauffolgenden Tag tatsächlich drankommt. Ausser bei Notfällen.	A
Programm	Kriterien	Das ist richtig. Am Beispiel des Karolinska-Spitals wird dies deutlich, wo man für gewisse Säle lange Operationen mit weniger Personaleinsatz (sowohl OP- und Anästhesiepersonal) durchführt und in anderen Sälen kurze Operationen mit	A

Abbildung 4.15: Vorlagen-File im Word

Die Aufgabe bestand nun darin, daraus eine gute Page zu erstellen. Gestützt

auf die Aussage, dass alle Wissenspunkte in allen Spitälern identisch sind[5], wurde ein Page-Generator in Java geschrieben.

Die Word-Dokumente mussten als erstes als HTML-Dokumente abgespeichert werden. Der Grund lieg darin, dass es einfacher ist, ein Programm zu schreiben, das HTML-Code analysiert als Word-Code, denn letzterer ist nirgends dokumentiert und ändert sich von Version zu Version.

Der Page-Generator selbst erzeugt als erstes die Menü- und danach die einzelnen Text-Files. Die Menüfiles gibt es doppelt; einmal für die rechte und einmal für die linke Seite. Die Text-Files gibt es nur einmal, da sie keinerlei Navigationselemente besitzen.

Die Menü-Files sind wie folgt benannt:

- Der erste Buchstabe entscheidet, ob es sich um ein linkes oder rechtes Menü handelt, wobei 'L' für links und 'R' für rechts steht.
- Der zweite Buchstabe ist fix ein 'm'
- Die nächste Buchstabengruppe beschreibt die Arbeitsgruppe (gemäss der letzten Spalte im Word-Dokument), gefolgt von einem '_':
  - A steht für Anästhesie
  - AP steht für Anästhesie Pflege
  - C steht für Chirurgie
  - LP steht für Lagerungsspezialist
  - P steht für OP-Pflegedienst
  - X steht für alle Berufsgruppen
- Nun folgt der ganze Name des Menüpunktes. Gibt es ein Submenü, wird nach einem erneuten '_' der Name des Submenüs eingefügt.

  Da es Probleme mit den Sonderzeichen geben kann, werden statt den Umlauten ä, ö und ü die Buchstaben a, o und u verwendet. Leerzeichen werden durch ein '-' ersetzt.
- als letztes folgt nun die Dateiendung '.html'

[5] Dies war leider nicht so. Einige Dokumente mussten erweitert werden, so dass wirklich jeder Punkt in jedem Dokument vorkam. Mehr dazu im Kapitel 4.5 auf Seite 20.

Lässt man die ersten zwei Zeichen des oben beschriebenen Namens weg, so erhält man die korrespondierende Textdatei.

Beispiel: `LmX_konzept-b_anzahl-sale.html` ist die linke Menü-Datei zum Punkt "Konzept B" und zum Unterpunkt "Anzahl Säle" für alle Berufsgruppen. Die entsprechende Text-Datei heisst `X_konzept-b_anzahl-sale.html`.

Die Unterscheidung nach Spitäler geschieht nicht im Dateinamen. Die Daten zu unterschiedlichen Spitälern befinden sich in unterschiedlichen Ordnern.

Wenn nun die Word-Dateien nicht konsistent sind, also nicht alle Wissenspunkte in jedem Spital vorhanden sind, wird ein Spitalwechsel zu einer nichtvorhandener Datei führen und mit einer Fehlermeldung des Browsers quittiert. Dies kann nicht durch JavaScript abgefangen werden. Nichtkonsistente Daten werden sich hier rächen. Wie man die Konsistenz sicherstellt, ist in 4.5 auf Seite 20 beschrieben.

4.5 gelöste infomatische Probleme

Java bring von Hause aus Sicherheitsanforderungen mit, welche zum Schutze des heimischen Computers gedacht sind. Dadurch ergeben sich aber auch Probleme. Ein konkretes Problem ist, dass man keine Dateien vom Computer des Anwenders / der Anwenderin lesen darf.[6] Nun müssen aber die Applets von MEDINO Bilder anzeigen, die sie von der CD lesen, was sie aber nicht dürfen. Dieser Widerspruch wurde gelöst, indem die Bilder in Ordner eingefügt wurden, die sich im Verzeichnisbaum unterhalb vom Applet-Ordner befinden, denn auf diese darf zugegriffen werden.

Ein weiteres Problem gab es bei den Wissenspunkten. Der Page-Generator generiert nur Dateien von einem Spital und weiss nicht, was die anderen Spitäler für Wissenspunkte haben. In der Navigation der Page muss aber sichergestellt werden, dass man von jedem beliebigen Wissens-Punkt eines Spitals auf denselbigen in einem anderen Spital springen kann. Dazu müssen alle Punkte in allen Spitäler vorhanden sein.

Der Generator kann dies nicht sicherstellen, da er spitalweise arbeitet. Somit muss dafür gesorgt werden, dass die ursprünglichen Daten diese Eigenschaft aufweisen.

[6] Da Applets auch im Internet laufen können, ist diese Einschränkung sehr wichtig, da man ansonsten die Festplatte(n) der Kundschaft ausspionieren und manipulieren könnte.

Im ursprünglichen Word-Dokument des Universitätsspitals gab es zu gewissen Punkten keine Einträge. Diese Punkte müssen aber vorhanden sein. Somit wurden solche Punkte eingefügt und mit dem Text `Keine Aussage zu diesem Thema` unter einer nichtexistenten Arbeitsgruppe (z.B.: `z`) eingefügt. Dadurch wird dieser Text nur angezeigt, wenn man alle Berufsgruppen ausgewählt hat.
Es kam auch vor, dass in gewissen Spitäler gewisse Berufgruppen überhaupt nicht befragt wurden. Damit aber die Navigation funktioniert, muss es in jeder Berufsgruppe mindestens einen Eintrag haben. Die Konsistenz wurde erreicht, indem in einem Wissenspunkt, zu dem niemand Stellung genommen hat, der Eintrag `Keine Aussage zu diesem Thema` nicht mit der Gruppe `z` sondern mit der entsprechenden Gruppe versehen wurde. (Dies war übrigens meistens der Punkt "Kommentar – allgemein".)

4.6 Ausblick

Auch wenn hier eine CD konzipiert worden ist, lässt sich MEDINO auch über einen Web-Server betreiben. Dies ist auch zu empfehlen, falls die Geschwindigkeit im Web für eine grosse Mehrheit genügend schnell ist.

MEDINO kann leicht ohne Programmierkenntnisse auf ein anderes Arbeitssystem adaptiert werden. Die MEDINO-Datenbank kann leicht mit Informationen über andere Arbeitssysteme gefüllt werden, auch wenn sich diese von der besonderen Situation eines Spitals unterscheiden. Die Dokumentation der Arbeitssysteme muss lediglich über neue Bilder und Texte geschehen. Die vorhandenen Programme von MEDINO müssen nicht angepasst werden.

Kapitel 5

Ergonomie-Prinzipien für Software

Der 10. Teil der ISO 9241-Norm (ISO 9241/10) ist eine internationale Ergonomienorm. Die Beurteilung von Softwaresystemen geschieht anhand der folgenden Kriterien:

- **Aufgabenangemessenheit:** Unterstützt die Software die Lösung der Arbeitsaufgabe? Wird man nicht unnötig belastet?
- **Selbstbeschreibungsfähigkeit:** Bekommt man genügend Informationen, wie das System funktioniert? Ist der Funktionsumfang ersichtlich? Hat es eine räumliche Orientierung durch konsistente Gestaltung?
- **Steuerbarkeit:** Kann man die Art und Weise, wie man mit der Software arbeitet, beeinflussen? Kann man das Arbeitstempo selbst wählen?
- **Erwartungskonformität:** Konsistenz in der Bedienung. Gestaltung in Übereinstimmung mit den Merkmalen der gewohnten Umgebung.
- **Fehlertoleranz:** Minimaler Korrekturaufwand bei Fehlbedienung. Fehlermeldungen sind sachlich, verständlich und konstruktiv.
- **Individualisierbarkeit:** Kann man die Software ohne grossen Aufwand auf die individuellen Bedürfnisse anpassen?
- **Lerneignung:** Einarbeitung ohne grossen Aufwand. Risikoarmes Ausprobieren. Ist Bekanntes übertragbar?

5.1 Aufgabenangemessenheit

MEDINO ist ein Informationstool. Somit muss man sich hier fragen, ob der/die BenutzerIn leicht an die gewünschte Information kommt.

Die Hauptseite bietet einem die Möglichkeit, in zwei Gebiete zu verzweigen. Will man etwas über ein Spital wissen, geschieht dies über die Spitalauswahl (Auswahl des Arbeitssystems), ansonsten entscheidet man sich für einen Punkt aus den Wissensthemen.
Innerhalb eines Spitals kann man sich für die Architektur oder die Arbeitsabläufe entscheiden.

Probleme könnten BenutzerInnen haben, die sich mit der Studie, auf die sich MEDINO bezieht, zu gut auskennen und die Arbeitsabläufe zum Konzept A haben wollen, die es aber in dieser Form nicht gibt. Die Verwirrung klärt sich aber rasch, da die Navigationstiefe sehr gering ist. Hat man sich für einen falschen Weg entschieden, merkt man dies spätestens nach zwei/drei Klicks und da die Verzweigungsmöglichkeiten von der Hauptseite aus zwei beträgt, gibt es also nur einen "falschen" Weg. Man dürfte also schnell zur gewünschten Information kommen.

5.2 Selbstbeschreibungsfähigkeit

Ist der/die BenutzerIn gewohnt mit dem Browser umzugehen, so dürften keine Probleme mit der Navigationsart bestehen. MEDINO bedient sich der Hyperlinktechnik.

Alle Seiten sind einheitlich gestaltet. Sie haben eine Überschrift. Unterhalb dieser Überschrift sind alle Möglichkeiten aufgeführt, weiter zu navigieren oder weitere Funktionen zu aktivieren. Man weiss also jederzeit, wo man sich befindet und was man von dieser Stelle aus tun kann.

Im Bereich Wissensthemen, der gegebenenfalls zu Navigationsprobleme führen könnte, kann der/die BenutzerIn Hilfen einblenden.

5.3 Steuerbarkeit

Der/Die BenutzerIn hat in den Wissensthemen eine direkte Möglichkeit die Art, wie die Informationen erscheinen, zu beeinflussen. Der Vergleichsmodus kann ein- oder ausgeschaltet werden.

Ansonsten ist die Steuerung vorgegeben, was für eine Informations-CD nicht verwunderlich ist. Welche Informationen aber erscheinen, obliegt der Kontrolle der BenutzerInnen.

Das Arbeitstempo ist nicht vorgegeben. Im Bereich Arbeitsabläufe hat man genügend Zeit, die Bilder anzuschauen und die Begleittexte zu lesen, da man selbst bestimmt, wann das nächste Bild kommt. Man kann sogar in der Zeitachse zurück gehen.

5.4 Erwartungskonformität

Wie bereits zuvor erwähnt, sind alle Seiten (bis auf die Hauptseite) einheitlich mit einer Überschrift versehen. Für die weitere Navigation sind unterhalb der Überschrift Links angebracht.
Die Links verhalten sich, wie es ein Internetbenutzer / eine Internetbebnutzerin von anderen Seiten gewöhnt ist.

Die Wissensthemen sind in Haupt- und Untermenüs organisiert. Auch diese Art der Navigation sollte unter den Benutzern / Benutzerinnen bekannt sein.

Einzig der Vergleichmodus könnte anfangs Orientierungs-Probleme bereiten. Um Bedienungsmissverständnisse zu vermeiden, gibt es die Möglichkeit, eine Hilfe einzublenden.

5.5 Fehlertoleranz

Fehlbedienungen werden als bewusste Auswahl gewertet. Die Korrektur muss also vom Benutzer / von der Benutzerin erfolgen. Dies geschieht durch die normale Navigation. Demzufolge ist die Fehlerkorrektur nicht aufwendiger als die Navigation an sich.

Nur beim Feedbackformular könnten Bedienungsformen zu internen Systemfehlern führen. Im Feedbackformular wird überprüft, ob der/die BenutzerIn alle geforderten Angeben getätigt hat und ob er/sie einen Wissenspunkt ausgewählt hat. Ist dies nicht der Fall, wird ein entsprechendes Hinweis-Fenster eingeblendet. Nachdem dieses vom User geschlossen wurde, befindet sich der Cursor im Feld, das den Fehler verursacht hat.

5.6 Individualisierbarkeit

Die Individualisierbarkeit gibt es einzig in den Wissensthemen, wo man sich zwischen dem Vergleichmodus "ein" oder "aus" entscheiden kann.

Ansonsten gilt es zu sagen, dass MEDINO die Einstellungen des Browser übernimmt, im Gegensatz zu anderen Seiten, welche Schriftgrösse und -art vorgeben. D.h. der/die BenutzerIn kann mit der gewohnten Schriftgrösse und -art, mit welcher er/sie am besten zurechtkommt, arbeiten. Einzig die Mindmap und deren Hilfetexte haben eine fixe Schriftgrösse, da sie als Bilder abgespeichert sind.

5.7 Lerneignung

Hier erweist sich die Wahl, MEDINO im Browser laufen zu lassen, als Vorteil. Die BenutzerInnen können ihre Surf-Kenntnisse voll einsetzen. Die Einarbeitungszeit ist sehr gering. Durch die übersichtliche und einheitliche Gestaltung der Seiten von MEDINO ist ein Ausprobieren risikolos.

Kapitel 6

unterstützte Browser

Die Entscheidung, MEDINO via Browser zu benutzen, stand bereits zu Beginn fest. Damit wollten wir erreichen, dass der/die BenutzerIn nichts installieren muss. MEDINO ist also sofort einsetzbar, unabhängig von der verwendeten Plattform. Heutzutage steht in praktisch allen Betriebssystemen ein Browser bereit.

Es wurde Wert gelegt, die verwendeten Hilfsmittel und Befehle so allgemein wie möglich zu halten und vermehrt auf Standards zu achten.

6.1 theoretisch unterstützte Browser

Der verwendete Browser muss mindestens den JavaScript-Standard der Version 1.1 und Java 1.1 unterstützen, sowie mit Cookies umgehen können.

Laut Angaben der entsprechenden Hersteller sind dies unter anderem

- Netscape: ab Version 3.0 und
- Internet-Explorer: ab Version 4.0

6.2 getestete Browser

Da Theorie und Praxis nicht immer übereinstimmen, wurde MEDINO unter folgenden Plattformen und mit folgenden Browsern getestet:

Betriebssystem	Browser	Resultat
Windows 95	Internet Explorer 4.0	O.K.
Windows 98	Internet Explorer 5.0	O.K.
Windows NT	Internet Explorer 5.0	O.K.
Windows 95	Netscape Navigator 4.7	O.K.
Windows 98	Netscape Navigator 4.7	O.K.
Windows NT	Netscape Navigator 4.7	O.K.
Solaris (Unix)	Netscape Navigator 4.08 / 4.5 / 4.6 / 4.7.3	O.K.
Solaris (Unix)	Netscape Navigator 4.04	Security-Probleme in JAVA. (Bereich Arbeitssysteme funktionsuntüchtig.)
Solaris (Unix)	Netscape Navigator 3.04	Security-Probleme in JAVA. JavaScript kennt den Befehl String.slice nicht. (Weder Arbeitssysteme noch Wissensthemen funktionieren.)
Macintosh	Internet Explorer 5.0	Fehlfunktion in den Wissensthemen.
Macintosh	Netscape Navigator 4.7	Fehlfunktion in den Wissensthemen und Security-Probleme in JAVA.

Die Test-CD wurde beim Test für Windows benutzt. Für Unix wurde das Zip-Archiv lokal entpackt und MEDINO via Festplatte getestet. Für den Macintosh wurde die MEDINO-Webpage benutzt.

6.3 mögliche Probleme im Betrieb

Trotz den Test mit den häufigsten Browsern und Betriebssystemen und trotz den Bemühungen, Befehle so allgemein wie möglich zu halten und vermehrt auf Standards zu achten, besteht die theoretische Gefahr, dass MEDINO auf bestimmten Browsern, auf bestimmten Betriebssystemen nicht oder nur teilweise funktionieren wird, sei es weil der verwendete Browser sich nicht nach dem Standard richtet oder schlicht fehlerhaft ist.

Ein mögliches Problem könnte man mit Java haben. Java-Applets unterliegen gewissen Sicherheitsanforderungen. Sie laufen in einem sogenannten Sandkasten ab. Bildlich gesprochen: In diesem Sandkasten können sie sich

austoben; darüber hinaus dürfen sie nichts. Damit wird verhindert, dass ein Applet eine Festplatte formatieren kann.

Was ein Applet darf oder nicht, bestimmt der Browser selbst. Ein Browserhersteller engt die Standardeinschränkungen von Sun[1] weiter ein. Somit kann er mehr verbieten, als notwendig. Der Browser bestimmt also, wie gross (besser: wie klein) der Sandkasten ist. Auch hier ist es möglich, dass bestimmte Browser auf bestimmten Betriebssystemen die Applets von MEDINO ausser Betrieb setzen.

[1] Java wurde von Sun entwickelt

Kapitel 7

Erweiterung von MEDINO (Anleitung)

7.1 Organisation der Files

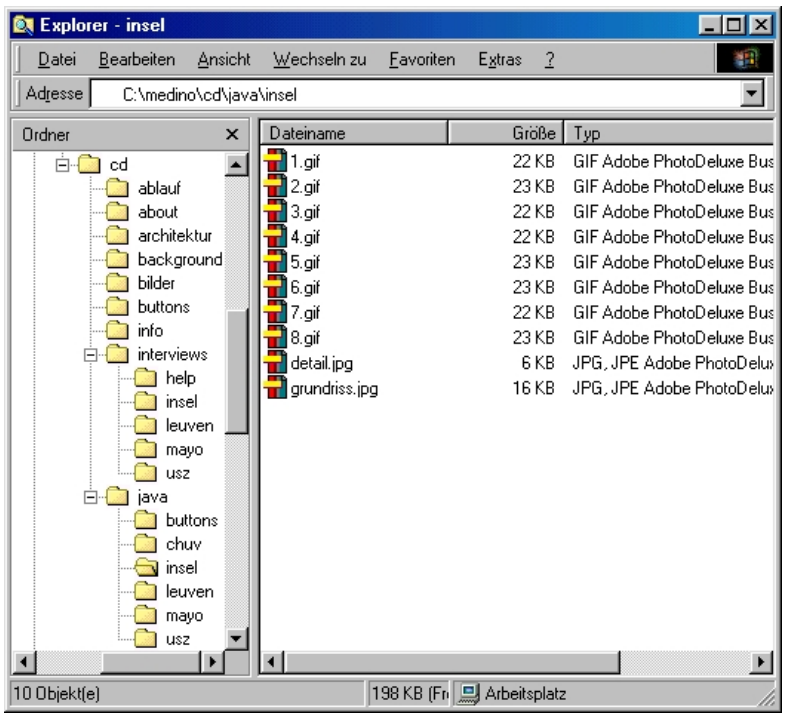


Abbildung 7.1: Organisation der Files in MEDINO

Die Abbildung 7.1 zeigt die Organisation der Files auf der CD (wobei der Ordner java/insel offen ist). Will man MEDINO erweitern, muss man sich mit dieser vertraut machen.
Falls einem die Struktur etwas mühsam erscheint, liegt es daran, dass es für Java-Applets gewisse Restriktionen gibt, die sich auf den Ablageort von Dateien auswirken. (Näheres gibt es im Kapitel 4.5 auf Seite 20.)

Für die Erweiterung sind folgende Ordner wichtig:

- ablauf: Hier befinden sich die HTML-Dateien der Spitäler betreffend der Arbeitsabläufe.
- architektur: Hier befinden sich die HTML-Dateien der Spitäler betreffend Architektur.
- buttons: Bilder, welche in der Spitalauswahl (Arbeitssysteme) benötigt werden.
- interviews: Enthält pro Spital einen Ordner mit den Daten zu den Wissensthemen. Im Ordner help befinden sich die Beschreibungstexte zur Mindmap. Daneben hat es weitere Files für die Navigation.
- java: Besitzt ebenfalls pro Spital einen Ordner, in welchem sich die Bilder zu den Abläufen und zur Architektur befinden.

7.2 Spitalauswahl (Wahl des Arbeitssystems)

Dies ist die linke Spalte der Hauptseite. Hierfür werden zwei Luftbilder benötigt. Das eine Bild wird angezeigt, wenn sich der Mauszeiger über dem Bild befindet, das andere, wenn sich der Mauszeiger nicht mehr darüber befindet.
Es wird empfohlen die Bilder *spital*.jpg und *spital*1.jpg zu benennen, wobei *spital* der in MEDINO verwendete Kürzel fürs Spital sein sollte.

Diese Bilder müssen in den Ordner `buttons` kopiert werden.

Das HTML-File `start.html` muss nun auch angepasst werden. Es befindet sich im Ordner `interviews`.[1]

Als erstes muss an die Zeilen

[1] Auch hier erschient auf den ersten Blick die Wahl des Ordners seltsam. Der Grund liegt darin, dass nur auf diese Weise das Zusammenspiel zwischen Mindmap und Vergleichsmodus via Cookies funktioniert.

```
  inselA = new Image(); inselA.src = "buttons/insel.jpg";
  inselB = new Image(); inselB.src = "buttons/insel1.jpg";
```

ähnliche Zeilen für das neue Spital angefügt werden. Hier werden die zwei Luftaufnahmen geladen. Mann muss beachten, dass man die richtigen Dateinamen verwendet.

Die folgenden Zeile müssen ebenfalls kopiert und korrekt angepasst werden. (Überall wo das Wort "insel" auftaucht, muss man das Kürzel des neuen Spitals einsetzen.[2])

```
<tr>
  <td align="CENTER" valign="TOP">
    <a href="architektur/insel.html" onMouseOver="window.
document.insel.src=inselB.src"
    onMouseOut="window.document.insel.src=inselA.src">
    <img name="insel" src="buttons/insel.jpg" width=190
      height=100 border=0 alt="Inselspital Bern" ><br>
    <font size="-1"><b>Inselspital Bern</b></font>
    </a>
  </td>
</tr>
```

7.3 Architektur eines Spitals

Ein Spital wird in der Architektur mit zwei Grundrissen vorgestellt. Einer für den OP-Trakt und einer für die Umgebung eines einzelnen OP-Saals[3]. Für das neue Spital erstellt man im Ordner `java` einen eigenen Ordner, den man geeignet benennet (z.B. durch den Kürzel des Spitals). Dorthin kopiert man diese zwei Bilder. Hier ist die Benennung der Bilder wichtig. Das Bild mit dem OP-Trakt muss `grundriss.jpg`, dasjenige mit dem OP-Saal muss `detail.jpg` heissen.[4]

Im Ordner `architektur` befinden sich die einzelnen HTML-Files der Spitäler. Es empfiehlt sich eines der Files (z.B. insel.html) zu kopieren und dann anzupassen. Für die Bearbeitung benötigt man Kenntnisse in HTML. Es muss

[2]Vorausgesetzt man hat sich an die zuvor vorgeschlagene Benennung gehalten.

[3]Dieses Bild sollte idealerweise wieder für die Abläufe benutzt werden. Somit kann der/die BenutzerIn sich besser orientieren.

[4]Alle Buchstaben müssen klein geschrieben werden, sonst findet das Java-Applet die Bilder nicht.

sichergestellt werden, dass die Links stimmen, die richtige Flugaufnahme gezeigt wird und dass die richtigen Texte verwendet werden.

Die Parameter mit welcher das Java-Applet aufgerufen wird, müssen auch angepasst werden.

In `<param name="spital" value="insel">` muss der Wert `insel` durch den Namen des Ordners ersetzt werden, in welchem sich die beiden Bilder `detail.jpg` und `grundriss.jpg` befinden.

`<param name="standard" value="...">` ist der Text, welcher eingeblendet wird, wenn sich die Maus über keiner sensitiven Stelle befindet.

Die beiden Parameter

```
<param name="gr1" value="50|40|106|88|OP6">
<param name="det1" value="6|3|97|93|OP4">
```

bestimmen einen sensitiven Bereich in einem Bild mitsamt der Erklärung, welche eingeblendet werden soll. Die Koordinaten und die Beschreibung werden durch senkrechte Striche getrennt.
`gr1` ist die erste sensitive Fläche im Bild `grundriss.jpg`. Man kann beliebig viele solche Flächen definieren, indem man weitere Parameter im Stile `gr`*Zahl* einfügt. Man muss einfach darauf achten, dass die Nummerierung nicht unterbrochen wird.
`det1` ist analog die erste sensitive Fläche im Bild `detai.jpg`.

7.4 Arbeitsablauf in einem Spital

Die Bilder, welche angezeigt werden sollen, müssen gemäss ihres Auftretens nummeriert werden. Verwendet man das gif-Format, heissen sie 1.gif, 2.gif, 3.gif und so weiter. Sie müssen nun in den Ordner kopiert, in welchem sich bereits `detail.jpg` und `grundriss.jpg` befinden. (siehe auch Abbildung 7.1)

Im Ordner `ablauf` befinden sich die einzelnen HTML-Files der Spitäler. Es empfiehlt sich eines der Files (z.B. insel.html) zu kopieren und dann anzupassen, wozu wiederum HTML-Kenntnisse benötigt werden.

Folgende Parameter müssen aufs neue Spital angepasst werden:

```
<param name="spital" value="insel">
<param name="anzahl_bilder" value="8">
<param name="format_ablaufbilder" value=".gif">
<param name="text1" value="Dies ist der 1. Text.">
```

Es wird der Name des Ordners benötigt, in welchem sich die Ablaufbilder befindet. Des weiteren wird die Anzahl der Bilder, die angezeigt werden sollen, als auch deren Format verlangt.[5]
Die Texte, welche zu den einzelnen Bilder angezeigt werden sollen, werden mit dem Parameter `text`*Nummer* übergeben.

7.5 Wissensthemen

Die Wissensthemen werden mit dem Programm makepage erweitert. Bevor man aber makepage startet, sollte man im Ordner interviews einen neuen Ordner erstellen (idealerweise mit dem Kürzel des Spitals benannt), in welchen die Files kommen.

Makepage befindet sich auf der Admin-CD im Ordner util/programme. Um es zu starten, benötigt man das JDK von Java, welches frei erhältlich ist.[6] Die Abbildung 7.2 zeigt eine erfolgreiche Ausführung von makepage.

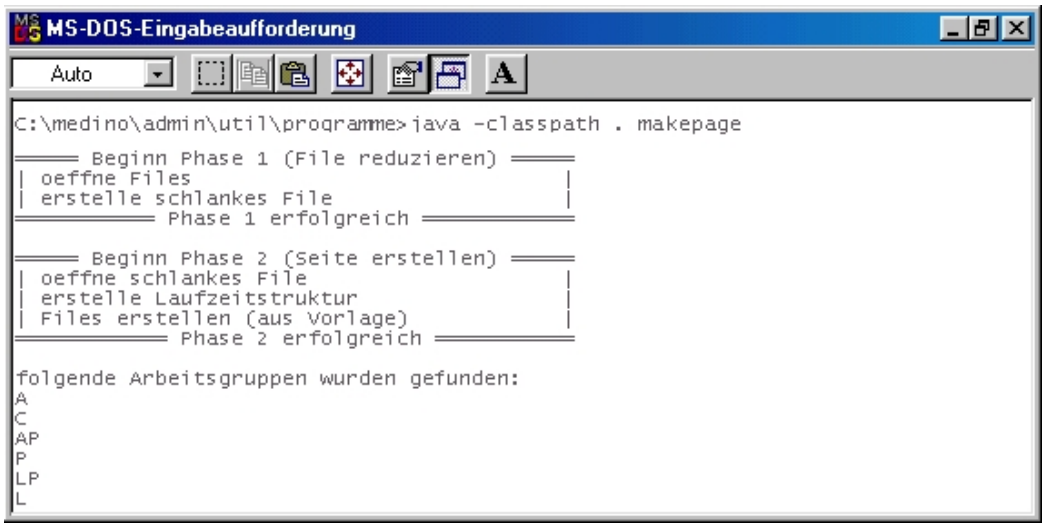


Abbildung 7.2: Erfolgreiche Ausführung von makepage

Hat man makepage gestartet, wird man in Fenster 7.3 aufgefordert, die Vorlagendatei[7] zu laden.

[5] Das Java-Applet kann gifs oder jpgs anzeigen.

[6] Auf der Admin-CD im Ordner jdk befindet sich das JDK 1.2.2 für Windows 95/98 oder Windows NT. Für andere Betriebssysteme gilt es die Internetseiten vom Sun (http://java.sun.com/products/jdk/1.2/) zu kontaktieren.

[7] Das Wordfile muss zuvor als HTML-Dokument abgespeichert worden sein.

Abbildung 7.3: Laden der Vorlagedatei in makepage

Die Dateien die makepage erstellt, sollte man in den zuvor erstellten Ordner kopieren. Dies geschieht im Fenster 7.4

Sollte das Vorlagenfile nicht korrekt gewesen sein, erfolgt eine Fehlermeldung im DOS-Prompt (oder UNIX-Shell). Wichtig ist, dass Inkonsistenten zwischen den Spitälern nicht erkannt werden können. Es ist die Angelegenheit derjenigen Person, welche die Interviewdaten erweitert, für Konsistenz zu sorgen. Siehe Kapitel 4.5 auf Seite 20.

Die einzige manuelle Änderung muss im Ordner interviews an den Files `Lselect.html` und `Rselect.html` durchgeführt werden. In beiden Files ist dies dieselbe Änderung. Die folgende Zeilen müssen durch eine weitere ergänzt werden. wobei der Wert hinter `value` der Name des Ordners ist, in welchen makepage die erstellten Files kopiert hat.

```
<option value="insel">Inselspital Bern</option>
<option value="usz">Universit&auml;stsspital Z&uuml;rich
</option>
<option value="leuven">Universit&auml;tsspital Leuven</option>
<option value="mayo">Mayo Medical Center</option>
```

Abbildung 7.4: Zieldatei sichern im Arbeitsordner von makepage

Anhang A

Glossar

Applet siehe Java-Applet.

Browser ist ein Programm, mit welchem man HTML-Seiten (Internet-Seiten) betrachten kann. Die verbreitetsten Browser sind der Internet-Explorer von Microsoft und der Netscape Navigator.

Cookies erlauben einem JavaScript-Programm, Daten auf der Festplatte des Anwenders / der Anwenderin zu speichern. Dies geschieht jedoch in kontrollierter Form. Ein "Ausspionieren" ist dabei ebenso wenig möglich wie das Platzieren von ausführbarem Code, also etwa Viren. Ein Cookie lässt sich mit einem Eintrag in einer Konfigurationsdatei vergleichen.

HTML bedeutet HyperText Markup Language. HTML ist eine sogenannte Auszeichnungssprache (Markup Language). Sie hat die Aufgabe, die logischen Bestandteile eines Dokuments zu beschreiben. Als Auszeichnungssprache enthält HTML daher Befehle zum Markieren typischer Elemente eines Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen.

Java ist eine von Sun Microsystems entwickelte, vollkommen plattformunabhängige Programmiersprache.

Java-Applets laufen im Anzeigefenster des java-fähigen Browsers ab. Java ist dazu geeignet, um Animationen, Simulationen, Echtzeitanwendungen sowie interaktive Anwendungen zu realisieren. Mit Java konnten die OP-Arbeitsabläufe und die Dokumentation der OP-Architekturen im Bereich Arbeitssysteme realisiert werden.

JavaScript ist kein direkter Bestandteil von HTML, sondern eine eigene Programmiersprache. Diese Sprache wurde jedoch eigens zu dem Zweck geschaffen, HTML-Autoren ein Werkzeug in die Hand zu geben, mit dessen Hilfe sich HTML-Seiten optimieren lassen. JavaScript wird auf der MEDINO-CD ledigich als Navigations-Hilfe verwendet. Um eine einfache Navigation zu ermöglichen, reicht HTML alleine nicht aus.

JSP steht für Java Server Pages. Wie es der Name sagt, läuft JSP auf einen Server (also nicht auf dem lokalen Rechner). Die Syntax ist identisch zu Java. Der Code lässt sich in HTML-Seiten einbauen.

Parser ist ein Programm, welches eine Folge von Zeichen nach gewünschten Informationen absucht. Der Page-Generator für die Wissensthemen ist solch ein Parser. Er parsed (analisiert) die Tabelle, in welcher sich die Daten befinden.

PHP ist die Abkürzung für "PHP: Hypertext Preprocessor" und ist eine Skriptsprache, die sich in HTML einbinden lässt. Das Ziel der Sprache ist es, das Schreiben von Programmen zur Erzeugung von dynamisch generierten Seiten zu erleichtern und zu beschleunigen. PHP kann nur auf Server laufen, also nicht auf dem lokalen Comuper.

Anhang B

Quellcodes

B.1 TextBox.java

Die Klasse TextBox ist zuständig für die Darstellung der Texte in den Architektur-Plänen und den Arbeits-Abläufen.

```
//-------------------------------------------------------------------------------
// Niko Kaintantzis                    "medino"                        17.04.2000
//                                   TextBox.java
//-------------------------------------------------------------------------------

import java.awt.FontMetrics;
import java.awt.Graphics;
import java.awt.Color;
import java.util.StringTokenizer;
import java.util.NoSuchElementException;


class TextBox{
  private Graphics g;
  private int x=0,y=0,w=0,h=0;
  private FontMetrics font;
  private Color textColor;


  public void paint(Graphics g,String txt,int x,int y,int w,FontMetrics font, Color tc){
    this.x=x;
    this.y=y;
    this.w=w;
    this.font=font;
    this.g=g;
    this.textColor=tc;
    repaint(txt);
  }

  public void repaint(String txt){
    if (txt != ""){
      int y=this.y+font.getAscent();
      String line="";
```

```
      String buffer;
      StringTokenizer st = new StringTokenizer(txt, " ");
      g.setColor(textColor);
      try{
        while (true) {
          buffer=st.nextToken();
          while(font.stringWidth(line+buffer+" ")<w){
            line+=buffer+" ";
            buffer=st.nextToken();
          }
          g.drawString(line,x,y);
          y+=font.getHeight() ;
          line=buffer+" ";
        }
      }
      catch (NoSuchElementException e){
        g.drawString(line,x,y);
        h=y+font.getHeight() ;
        return;
      }
    }
  }

}
```

B.2 ablauf.java

Das Applet ablauf stellt die Arbeitsabläufe dar.

```
//-------------------------------------------------------------------------------
// Niko Kaintantzis                    "medino"                        15.04.2000
//                                   ablauf.java
//-------------------------------------------------------------------------------

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Color;
import java.awt.Rectangle;
import java.awt.FontMetrics;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.Vector;
import java.util.NoSuchElementException;

public class ablauf extends java.applet.Applet {
  private Vector imgs= new Vector(), texts= new Vector();
  private Color silver = new Color(12632256);
  private Image img, prev, next;
  private int iWidth, iHeight, cur_img, TxtWidth, anz;
  private String BNr,text,format;
  private FontMetrics font;
  private Rectangle prev_r, next_r;
  private TextBox textBox=new TextBox();

  public void init() {
    // Parameter einlesen
    String spital = getParameter("spital");
```

```
    anz = (new Integer(getParameter("anzahl_bilder"))).intValue();
    format = getParameter("format_ablaufbilder");
    int i;
    for (i=1; i<=anz; i++){
      img = getImage(getCodeBase(),"./" +spital+ "/" +i+format);
      imgs.addElement(img);
      text = getParameter("text"+i);
      texts.addElement(text);
    }
    // Startdarstellung einrichten
    img=(Image)imgs.elementAt(0);
    cur_img=1;
    BNr=cur_img+". Bild von "+anz;
    text=(String)texts.elementAt(0);
    prev = getImage(getCodeBase(),"./buttons/prev.gif");
    next = getImage(getCodeBase(),"./buttons/next.gif");
    this.addMouseListener(new Listener());

  }

  public void paint(Graphics g) {
    int prev_x_pos, next_x_pos, prev_x_width, next_x_width, button_height;
    font=g.getFontMetrics();
    g.setColor(Color.white);
    // Applet-Hintergrund weiss fuellen
    g.fillRect(0,0,1000,1000);
    g.setColor(Color.black);
    g.drawImage(img, 5+5, 5+5, this);
    iWidth = img.getWidth(this);
    iHeight = img.getHeight(this);
    // Bild umranden
    g.drawRoundRect(5,5,iWidth+10,iHeight+10,10,10);
    g.setColor(silver);
    // Navigationsflaeche grau faerben und Elemente einfuegen
    g.fillRoundRect(5,iHeight+15+5,iWidth+10,font.getHeight()+10,10,10);
    g.setColor(Color.black);
    TxtWidth = 5+ font.stringWidth(BNr);

    g.drawString(BNr, (iWidth+5+10-TxtWidth)/2, iHeight+20+5+font.getAscent());
    g.setColor(Color.gray);
    prev_x_pos=5+3;
    prev_x_width=(iWidth+5+10-TxtWidth)/2-2*3-10;
    next_x_pos=5+3+(iWidth+5+10-TxtWidth)/2-2*3+TxtWidth;
    next_x_width=iWidth+5+10-next_x_pos-3;
    button_height=font.getHeight()+10-2*3;
    g.fillRoundRect(prev_x_pos,iHeight+15+5+3,prev_x_width,button_height,10,10);
    g.fillRoundRect(next_x_pos,iHeight+15+5+3,next_x_width,button_height,10,10);
    g.drawImage(prev,prev_x_pos+(prev_x_width-prev.getWidth(this))/2,iHeight+20+5, this);
    g.drawImage(next,next_x_pos+(next_x_width-next.getWidth(this))/2,iHeight+20+5, this);
    prev_r=new Rectangle(prev_x_pos,iHeight+20+5,prev_x_width,button_height);
    next_r=new Rectangle(next_x_pos,iHeight+20+5,next_x_width,button_height);
    // Text-Feld umranden und Text einfuegen
    g.drawRoundRect(iWidth+15+5,0+5,210,iHeight+15+font.getHeight()+10,10,10);
    textBox.paint(g,text,iWidth+15+5+5,0+5+5,200,font,Color.black);
  }


  //inner class
  class Listener extends MouseAdapter {
    private Rectangle lastrect;

    public void mousePressed(MouseEvent e) {
```

```
      Rectangle r = findrect(e);
      if (r == null) return;
      Graphics g = Applet.this.getGraphics();
      g.setXORMode(Color.blue);
      g.drawRect(r.x, r.y, r.width, r.height);
      lastrect = r;
    }

    public void mouseReleased(MouseEvent e) {
      if (lastrect != null) {
        Graphics g = Applet.this.getGraphics();
        //g.setXORMode(Color.gray);
        //g.drawRect(lastrect.x, lastrect.y, lastrect.width, lastrect.height);
        Rectangle r = findrect(e);
        if ((r != null) && (r == lastrect)){ //wenn noch im gleichen Button
          if (r==prev_r && cur_img>1) cur_img--;
          if (r==next_r && cur_img==anz) cur_img=0;
          if (r==next_r && cur_img<anz) cur_img++;
          img=(Image)imgs.elementAt(cur_img-1);
          text=(String)texts.elementAt(cur_img-1);
          BNr=cur_img+". Bild von "+anz;
          repaint();
        }
        lastrect = null;
      }
    }

    protected Rectangle findrect(MouseEvent e) {
      int x = e.getX(), y = e.getY();
      if (prev_r.contains(x, y)) return prev_r;
      if (next_r.contains(x, y)) return next_r;
      return null;
    }
  }//Ende inner class
}
```

B.3 intro.java

Das Applet intro macht die Architekturpläne sensitiv.

```
//-------------------------------------------------------------------------------
// Niko Kaintantzis                    "medino"                        13.05.2000
//                                    intro.java
//-------------------------------------------------------------------------------

import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Rectangle;
import java.awt.FontMetrics;
import java.awt.Color;
import java.awt.event.MouseMotionAdapter;
import java.awt.event.MouseEvent;
import java.util.StringTokenizer;
import java.util.Vector;
import java.util.NoSuchElementException;
import java.lang.NumberFormatException;
```

```
public class intro extends java.applet.Applet {
  private Vector rects= new Vector();
  private Image gr, det;
  private int anz, gr_Width, gr_Height, det_Width, det_Height;
  private String text, standard;
  private TextBox textBox=new TextBox();

  public void init() {
    String spital = getParameter("spital");
    standard = getParameter("standard");
    text = standard;
    TextRectangle r;
    gr = getImage(getCodeBase(),"./" +spital+ "/grundriss.jpg");
    det = getImage(getCodeBase(),"./" +spital+ "/detail.jpg");
    for (int i=1; (r=getRectangleParameter("gr"+i,0)) !=null; i++ )
      rects.addElement(r);
    for (int i=1; (r=getRectangleParameter("det"+i,1)) !=null; i++ )
      rects.addElement(r);
    //System.out.println(rects.size());
    this.addMouseMotionListener(new Listener());
  }

  protected TextRectangle getRectangleParameter(String name, int offset){
    int x1, y1, x2, y2;
    String txt, value=getParameter(name);
    if (value == null) return null;
    try{
      StringTokenizer st= new StringTokenizer(value, "|");
      x1=Integer.parseInt(st.nextToken());
      y1=Integer.parseInt(st.nextToken());
      x2=Integer.parseInt(st.nextToken());
      y2=Integer.parseInt(st.nextToken());
      txt=st.nextToken();
    }
    catch (NoSuchElementException e) {return null;}
    catch (NumberFormatException e) {return null;}
    return(new TextRectangle(x1,y1,x2-x1,y2-y1,txt,offset));
  }

  public void paint(Graphics g) {
    gr_Width = gr.getWidth(this);
    gr_Height = gr.getHeight(this);
    det_Width = det.getWidth(this);
    det_Height = det.getHeight(this);
    g.setColor(Color.white);
    g.fillRect(0,0,1000,1000);
    g.setColor(Color.black);
    g.drawImage(gr,0,0, this);
    g.drawImage(det,0,gr_Height+15, this);
    g.drawRoundRect(det_Width+15,gr_Height+15,gr_Width-det_Width-15,det_Height,10,10);
    textBox.paint(g,text,det_Width+20,gr_Height+20,gr_Width-det_Width-15,
    g.getFontMetrics(),Color.black);
  }

  //inner classes
  static class TextRectangle extends Rectangle {
    private String txt;
    private int offset;

    public TextRectangle(int x, int y ,int w, int h, String txt, int offset){
      super(x,y,w,h);
      this.txt = txt;
```

```
      this.offset=offset;
    }

    public String Txt(){return this.txt;}

    public int Offset() {return offset;}
  }


  class Listener extends MouseMotionAdapter {

    private TextRectangle last=null;

    public void mouseMoved(MouseEvent e) {
      TextRectangle r = findrect(e);
      if (r == last) return;
      if (r == null){
        text=standard;
        last=null;
      }else{
        text=r.Txt();
        last=r;
      }
      repaint();
    }


    protected TextRectangle findrect(MouseEvent e) {
      int i, y, x = e.getX(), ey = e.getY();
      TextRectangle r;
      for (i=0; i<rects.size();i++){
        r=(TextRectangle)rects.elementAt(i);
        y=ey-r.Offset()*(gr_Height+15);
        //Offset() gibt 1 (unten) oder 0 (oben) zurueck
        if (r.contains(x,y)) return r;
      }
      return null;
    }
  }//Ende inner class
}
```

B.4 welcome.java

Dieses Applet wird in der Willkommensseite benutzt. Es hat keine Funkion. Dadurch wird lediglich erreicht, dass die Java Virtual Machine des Browser gestartet wird.

```
//-----------------------------------------------------------------------------
// Niko Kaintantzis                    "medino"                        13.05.2000
//                                   welcome.java
//-----------------------------------------------------------------------------

public class welcome extends java.applet.Applet {
}
```

B.5 makepage.java

makepage.java umfasst alle Klassen die benötigt werden, um aus dem Vorlagenfile die benötigten Pages zu den Wissensthemen zu generieren.

```
//-------------------------------------------------------------------------------
// Niko Kaintantzis                    "medino"                        13.05.2000
//                                   makepage.java
//-------------------------------------------------------------------------------

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.FileNotFoundException;
import java.io.IOException;

import java.util.Vector;
import java.util.Enumeration;

import java.awt.FileDialog;
import java.awt.Frame;



class Reducer{
  private BufferedReader in=null;
  private BufferedWriter out=null;
  private String in_file , out_file;

  public Reducer(String in_file, String out_file){
    this.in_file=in_file;
    this.out_file=out_file;
  }

  public boolean openFiles(){
    try {
      in = new BufferedReader(new FileReader(in_file));
    } catch (FileNotFoundException e) {
      System.out.println(" >> Fehler beim Lesen");
      return false;
    }
    try {
      out = new BufferedWriter(new FileWriter(out_file));
    } catch (IOException e) {
      System.out.println(" >> Fehler beim Schreiben");
      return false;
    }
    return true;
  }

  public boolean reduce(){
    String cache;
    int index;
    // suche Tabellenanfang (erste Zelle)
    try{
      cache=in.readLine();
      while ((index=cache.toLowerCase().indexOf("<td"))==-1) cache=in.readLine();
      cache=cache.substring(index);
```

```
      while ((index=cache.indexOf(">"))==-1) cache=in.readLine();
      cache=cache.substring(index+1);
      if (cache.length()==0) cache=in.readLine();
      // Tabellenanfang übertragen
      out.write("<table border=1>\n<tr>\n\t<td>\n");

      // Text suchen und übertragen
      while (true){
        while (cache.indexOf(" ")==0) cache=cache.substring(1);
        index=cache.indexOf("<");
        if (index==-1){ //ganze Zeile besteht aus Text
          out.write(cache);
          out.write(" ");
          cache=in.readLine();
        }else if (index!=0){ //Text gefunden
          out.write(cache.substring(0,index));
          out.write(" ");
          cache=cache.substring(index);
        }else if (index==0){ //wieder ein tag
          if (cache.toLowerCase().indexOf("<tr")==0){
            out.write("<tr>\n");
            while ((index=cache.indexOf(">"))==-1) cache=in.readLine();
            cache=cache.substring(index+1);
            if (cache.length()==0) cache=in.readLine();
          }else if (cache.toLowerCase().indexOf("</tr>")==0){
            out.write("</tr>\n");
            cache=cache.substring(5);
          }else if (cache.toLowerCase().indexOf("<td")==0){
            out.write("\t<td>\n");
            while ((index=cache.indexOf(">"))==-1) cache=in.readLine();
            cache=cache.substring(index+1);
            if (cache.length()==0) cache=in.readLine();
          }else if (cache.toLowerCase().indexOf("</td>")==0){
            out.write("\n\t</td>\n");
            cache=cache.substring(5);
          }else if (cache.toLowerCase().indexOf("</table>")==0){
            out.write("</table>");
            in.close();
            out.close();
            return true;
          }else{
            while ((index=cache.indexOf(">"))==-1) cache=in.readLine();
            cache=cache.substring(index+1);
          }
        }
        if (cache.length()==0) cache=in.readLine();
      }
    } catch (IOException e) { //Ende des Files erreicht
      System.out.println(" >> Fehler: File-Struktur der Vorlage nicht korrekt! ");
      return false;
    }
  }
}

class Organizer{
  private BufferedReader in=null;
  private BufferedWriter out=null;
  private String dir;
  private Vector groups=new Vector();

  public Organizer(String dir, String file){
    this.dir=dir;
```

```
  try {
    in = new BufferedReader(new FileReader(dir.concat(file)));
  } catch (FileNotFoundException e) {
    System.out.println(" >> Fehler beim Lesen der eigenen Vorlage");
    System.exit(1);
  }
}

public Enumeration elements() {return groups.elements();}

private void add(String name, String menu, String submenu, String txt){
  Group group=new Group(name);
  groups.addElement(group);
  group.insert(menu,submenu,txt);
}

private void insert(String group, String menu,String submenu, String txt){
  int gr_nr,index;
  if ((index=group.indexOf(" ")) >0 )
    group=group.substring(0,index);
  while (menu.endsWith(" ")) menu=menu.substring(0,menu.length()-1);
  while (submenu.endsWith(" ")) submenu=submenu.substring(0,submenu.length()-1);
  if (submenu.indexOf(" ")==0) submenu="";
  if ((gr_nr=getPos(group))==-1)
    this.add(group,menu,submenu,txt);
  else
    ((Group)groups.elementAt(gr_nr)).insert(menu,submenu,txt);
}

private int getPos(String name){
      for (int i=0; i<groups.size(); i++)
    if ( ((Group)groups.elementAt(i)).getName().equals(name) ) return i;
  return -1;
}

public void read(){
  String cache, menu, submenu, txt, group;
  try{
    in.readLine(); // <table border=1>
    cache=in.readLine();
    while (cache.indexOf("<tr>")!=-1){ //sonst </table>
      in.readLine(); // <td>
      menu=in.readLine();
      in.readLine(); //</td>
      in.readLine(); //<td>
      submenu=in.readLine();
      in.readLine(); //</td>
      in.readLine(); //<td>
      txt=in.readLine();
      in.readLine(); //</td>
      in.readLine(); //<td>
      group=in.readLine();
      in.readLine(); //</td>
      in.readLine(); //</tr>
      cache=in.readLine(); //<tr> oder </table>
      this.insert("X",menu,submenu,txt);
      this.insert(group,menu,submenu,txt);
    }
    in.close();
  } catch (IOException e) { //Ende des Files erreicht
    System.out.println(" >> Fehler: beim Parsen der eigenen Vorlage! ");
    System.exit(1);
```

```
    }
  }

  public void write(){
    Group x= (Group)groups.remove(0);
    Group g;
    Enumeration enum=groups.elements();
    x.writeMenus(dir,"L");
    x.writeMenus(dir,"R");
    x.writeText(dir);
    x.createMenus(groups, dir, "L");
    x.createMenus(groups, dir, "R");
    while (enum.hasMoreElements()){
      g=(Group)enum.nextElement();
      g.writeMenus(dir,"L");
      g.writeMenus(dir,"R");
      g.writeText(dir);
    }
  }


}

class Group{
  private String name;
  private Vector menus=new Vector();

  public Group(String name){
    this.name=name;
  }

  public void createMenus(Vector groups, String dir, String framepos){
    BufferedWriter out=null;
    Enumeration gr_enum, m_enum, sm_enum;
    String mfile_gr, mfile_menu, mfile_submenu, tfile_gr, tfile_menu, tfile_submenu;
    Group g;
    Menu menu,submenu;
    gr_enum=groups.elements();
    while (gr_enum.hasMoreElements()){
      g=(Group)gr_enum.nextElement();
      mfile_gr=framepos.concat("m").concat(g.getName()).concat("_");
      tfile_gr=g.getName().concat("_");
      m_enum=menus.elements();
      while (m_enum.hasMoreElements()){
        menu=(Menu)m_enum.nextElement();
        mfile_menu=mfile_gr.concat(menu.getURL());
        tfile_menu=tfile_gr.concat(menu.getURL());
        try {
          out = new BufferedWriter(new FileWriter(dir.concat(mfile_menu).concat(".html")));
          out.write("<HTML>\n<HEAD>\n<meta http-equiv=\"refresh\" content=\"0; URL=");
          out.write(framepos.concat("m").concat(g.getName()).concat(".html"));
          out.write("\">\n</HEAD>\n</HTML>");
          out.close();
        } catch (IOException e) {
          System.out.println(" >> Fehler beim Schreiben");
          System.exit(1);
        }
        try {
          out = new BufferedWriter(new FileWriter(dir.concat(tfile_menu).concat(".html")));
          out.write("<HTML>\n<head></head>\n<body bgcolor=\"white\">\n");
          out.write("Kein Eintrag\n</body>\n</HTML>\n");
          out.close();
```

```
      } catch (IOException e) {
        System.out.println(" >> Fehler beim Schreiben");
        System.exit(1);
      }
      if (menu.hasSubMenus()){
        sm_enum=menu.getSubmenus().elements();
        while (sm_enum.hasMoreElements()){
          submenu=(Menu)sm_enum.nextElement();
          mfile_submenu=mfile_menu.concat("_").concat(submenu.getURL()).concat(".html");
          tfile_submenu=tfile_menu.concat("_").concat(submenu.getURL()).concat(".html");
          try {
            out = new BufferedWriter(new FileWriter(dir.concat(mfile_submenu)));
            out.write("<HTML>\n<HEAD>\n<meta http-equiv=\"refresh\" content=\"0; URL=");
            out.write(mfile_menu.concat(".html"));
            out.write("\">\n</HEAD>\n</HTML>");
            out.close();
          } catch (IOException e) {
            System.out.println(" >> Fehler beim Schreiben");
            System.exit(1);
          }
          try {
            out = new BufferedWriter(new FileWriter(dir.concat(tfile_submenu)));
            out.write("<HTML>\n<head></head>\n<body bgcolor=\"white\">\n");
            out.write("Kein Eintrag\n</body>\n</HTML>\n");
            out.close();
          } catch (IOException e) {
            System.out.println(" >> Fehler beim Schreiben");
            System.exit(1);
          }
        }
      }
    }
  }
}

public void writeMenus(String dir, String framepos){
  Enumeration enum;
  BufferedWriter out=null;
  Menu menu,submenu;
  String head, body="" , tail;
  String prefixm,prefixt,submenu_link,menu_link, filename="error.html";
  int loops;
  head="<html>\n<head>\n";
  head=head.concat("\t<script type=\"text/javascript\" language=\"JavaScript\">\n");
  head=head.concat("\t<!--\n\tfunction Framewechsel(URL1,URL2)\n");
  head=head.concat("\t{\n\tparent.").concat(framepos);
  head=head.concat("menu.location.href=URL1;\n");
  head=head.concat("\tparent.").concat(framepos).concat("text.location.href=URL2;\n");
  head=head.concat("\t}\n\t//-->\n\t</script>\n");
  head=head.concat("</head>\n\n");
  head=head.concat("<body text=\"blue\" link=\"blue\" vlink=\"blue\"");
  head=head.concat("alink=\"red\" bgcolor=\"FFBB00\">\n");
  tail="</body>\n</html>";
  prefixm=framepos.concat("m").concat(name);

  // unselektierte Menuleiste erstellen
  try {
    out = new BufferedWriter(new FileWriter(dir.concat(prefixm).concat(".html")));
    out.write(head);
    out.write("<a name=\"this\"> </a>\n");
    enum=menus.elements();
    prefixm=prefixm.concat("_");
```

```
    while(enum.hasMoreElements()){
      menu=(Menu)enum.nextElement();
      menu_link=prefixm.concat(menu.getURL());
      prefixt=name.concat("_").concat(menu.getURL());
      out.write("<a href=\"javascript:Framewechsel('".concat(menu_link));
      if (menu.hasSubMenus())
        out.write(".html#this','../leer.html')\"");
      else
        out.write(".html#this','".concat(prefixt).concat(".html')\""));
      out.write(" style=\"text-decoration:none\">\n");
      out.write("\t<nobr>".concat(menu.getName()).concat("<br></nobr>\n"));
      out.write("</a>\n");
    }
    out.write(tail);
    out.close();
  } catch (IOException e) {
    System.out.println(" >> Fehler beim Schreiben");
    System.exit(1);
  }

  //menus erstellen
  for (int i=0; i<menus.size();i++){
    enum=menus.elements();
    body="";
    menu=(Menu)enum.nextElement();
    //Menus davor
    for( int j=0; j<i; j++){
      menu_link=prefixm.concat(menu.getURL());
      prefixt=name.concat("_").concat(menu.getURL());
      body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
      if (menu.hasSubMenus())
        body=body.concat(".html#this','../leer.html')\"");
      else
        body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
      body=body.concat(" target=\"").concat(framepos).concat("menu\"");
      body=body.concat(" style=\"text-decoration:none\">\n");
      body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
      body=body.concat("</a>\n");
      menu=(Menu)enum.nextElement();
    }
    //aktuelles Menu
    prefixt=name.concat("_").concat(menu.getURL());
    menu_link=prefixm.concat(menu.getURL());
    filename=menu_link.concat(".html");
    body=body.concat("<a name=\"this\"> </a>\n");
    body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
    if (menu.hasSubMenus())
      body=body.concat(".html#this','../leer.html')\"");
    else
      body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
    body=body.concat(" style=\"text-decoration:none; color: Red;\">\n");
    body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
    body=body.concat("</a>\n");
    //falls es keine Submenus gibt, wird schleife nicht genommen
    for (int k=0; k<((Menu)menus.elementAt(i)).getNofSubmenus();k++){
      submenu=((Menu)menus.elementAt(i)).getSubmenu(k);
      submenu_link=prefixm.concat(menu.getURL()).concat("_").concat(submenu.getURL());
      body=body.concat("<a href=\"javascript:Framewechsel('").concat(submenu_link);
      body=body.concat(".html#this','");
      submenu_link=prefixt.concat("_").concat(submenu.getURL());
      body=body.concat(submenu_link).concat(".html')\"");
      body=body.concat(" style=\"text-decoration:none\">\n");
```

```
      body=body.concat("\t\t<nobr>    ");
      body=body.concat(submenu.getName()).concat("<br></nobr>\n");
      body=body.concat("\t</a>\n");
    }
    //Menus danach
    if (enum.hasMoreElements()) menu=(Menu)enum.nextElement();
    for(int j=i+1; j<menus.size();j++){
      menu_link=prefixm.concat(menu.getURL());
      prefixt=name.concat("_").concat(menu.getURL());
      body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
      if (menu.hasSubMenus())
        body=body.concat(".html#this','../leer.html')\"");
      else
        body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
      body=body.concat(" target=\"").concat(framepos).concat("menu\"");
      body=body.concat(" style=\"text-decoration:none\">\n");
      body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
      body=body.concat("</a>\n");
      if (enum.hasMoreElements()) menu=(Menu)enum.nextElement();
    }
    try {
      out = new BufferedWriter(new FileWriter(dir.concat(filename)));
      out.write(head.concat(body).concat(tail));
      out.close();
    } catch (IOException e) {
      System.out.println(" >> Fehler beim Schreiben");
      System.exit(1);
    }
  }
  // falls es Submenus gibt, werden die Menufiles fuer diese erstellt.
  for (int i=0; i<menus.size();i++){
    loops=((Menu)menus.elementAt(i)).getNofSubmenus();
    for( int m=0; m<loops; m++ ){
      enum=menus.elements();
      body="";
      menu=(Menu)enum.nextElement();
      //Menus davor
      for( int j=0; j<i; j++){
        menu_link=prefixm.concat(menu.getURL());
        prefixt=name.concat("_").concat(menu.getURL());
        body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
        if (menu.hasSubMenus())
          body=body.concat(".html#this','../leer.html')\"");
        else
          body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
        body=body.concat(" target=\"").concat(framepos).concat("menu\"");
        body=body.concat(" target=\"").concat(framepos).concat("menu\"");
        body=body.concat(" style=\"text-decoration:none\">\n");
        body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
        body=body.concat("</a>\n");
        menu=(Menu)enum.nextElement();
      }
      //aktuelles Menu
      menu_link=prefixm.concat(menu.getURL());
      prefixt=name.concat("_").concat(menu.getURL());
      body=body.concat("<a name=\"this\"> </a>\n");
      body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
      if (menu.hasSubMenus())
        body=body.concat(".html#this','../leer.html')\"");
      else
        body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
      body=body.concat(" style=\"text-decoration:none; color: Red;\">\n");
```

```
body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
body=body.concat("</a>\n");
//submenus davor
for (int k=0; k<m; k++){
  submenu=((Menu)menus.elementAt(i)).getSubmenu(k);
  submenu_link=prefixm.concat(menu.getURL()).concat("_").concat(submenu.getURL());
  body=body.concat("\t<a href=\"javascript:Framewechsel('").concat(submenu_link);
  body=body.concat(".html#this','");
  submenu_link=prefixt.concat("_").concat(submenu.getURL());
  body=body.concat(submenu_link).concat(".html')\"");
  body=body.concat(" style=\"text-decoration:none\">\n");
  body=body.concat("\t\t<nobr>    ");
  body=body.concat(submenu.getName()).concat("<br></nobr>\n");
  body=body.concat("\t</a>\n");
}
// dieses Submenu
submenu=((Menu)menus.elementAt(i)).getSubmenu(m);
submenu_link=prefixm.concat(menu.getURL()).concat("_").concat(submenu.getURL());
body=body.concat("\t<a href=\"javascript:Framewechsel('").concat(submenu_link);
body=body.concat(".html#this','");
submenu_link=prefixt.concat("_").concat(submenu.getURL());
body=body.concat(submenu_link).concat(".html')\"");
body=body.concat(" style=\"text-decoration:none; color: Red;\">\n");
body=body.concat("\t\t<nobr>    ");
body=body.concat(submenu.getName()).concat("<br></nobr>\n");
body=body.concat("\t</a>\n");
filename=menu_link.concat("_").concat(submenu.getURL()).concat(".html");
// submenus danach
for (int k=m+1; k<((Menu)menus.elementAt(i)).getNofSubmenus(); k++){
  submenu=((Menu)menus.elementAt(i)).getSubmenu(k);
  submenu_link=prefixm.concat(menu.getURL()).concat("_").concat(submenu.getURL());
  body=body.concat("\t<a href=\"javascript:Framewechsel('").concat(submenu_link);
  body=body.concat(".html#this','");
  submenu_link=prefixt.concat("_").concat(submenu.getURL());
  body=body.concat(submenu_link).concat(".html')\"");
  body=body.concat("\t<a href=\"").concat(submenu_link).concat(".html\"");
  body=body.concat(" style=\"text-decoration:none\">\n");
  body=body.concat("\t\t<nobr>    ");
  body=body.concat(submenu.getName()).concat("<br></nobr>\n");
  body=body.concat("\t</a>\n");
}
//Menus danach
if (enum.hasMoreElements()) menu=(Menu)enum.nextElement();
for(int j=i+1; j<menus.size();j++){
  menu_link=prefixm.concat(menu.getURL());
  prefixt=name.concat("_").concat(menu.getURL());
  body=body.concat("<a href=\"javascript:Framewechsel('").concat(menu_link);
  if (menu.hasSubMenus())
    body=body.concat(".html#this','../leer.html')\"");
  else
    body=body.concat(".html#this','").concat(prefixt).concat(".html')\"");
  body=body.concat(" target=\"").concat(framepos).concat("menu\"");
  body=body.concat(" style=\"text-decoration:none\">\n");
  body=body.concat("\t<nobr>").concat(menu.getName()).concat("<br></nobr>\n");
  body=body.concat("</a>\n");
  if (enum.hasMoreElements()) menu=(Menu)enum.nextElement();
}
try {
  out = new BufferedWriter(new FileWriter(dir.concat(filename)));
  out.write(head.concat(body).concat(tail));
  out.close();
} catch (IOException e) {
```

```
          System.out.println(" >> Fehler beim Schreiben");
          System.exit(1);
        }
      }
    }
  }

  public void writeText(String dir){
    Enumeration enum=menus.elements();
    Menu menu;
    while (enum.hasMoreElements()){
      menu=(Menu)enum.nextElement();
      menu.write(dir, name.concat("_"));
    }
  }

  public void insert(String menu, String submenu, String txt){
    int pos=getPos(menu);
    if (pos==-1)
      menus.addElement(new Menu(menu,submenu,txt));
    else
      ((Menu)menus.elementAt(pos)).insert(submenu,txt);
  }

  private int getPos(String name){
        for (int i=0; i<menus.size(); i++)
      if ( ((Menu)menus.elementAt(i)).getName().equals(name) ) return i;
    return -1;
  }

  public String getName() {return name;}

}

class Menu{
  private String name;
  private Vector vect=new Vector();
  private boolean hasSubMenus=true;

  public Menu(String menu, String submenu, String txt){
    name=menu;
    if (submenu==""){
      vect.addElement(txt);
      hasSubMenus=false;
    }else{
      vect.addElement(new Menu(submenu,"",txt));
    }
  }

  public void write(String dir, String file_start){
    BufferedWriter out=null;
    Enumeration enum=vect.elements();
    Menu menu;
    String txt;
    if (hasSubMenus){
      while (enum.hasMoreElements()){
        menu=(Menu)enum.nextElement();
        menu.write(dir,file_start.concat(getURL()).concat("_"));
      }
    }else{
      try {
        out = new BufferedWriter(new FileWriter(dir.concat(file_start).
```

```
        concat(getURL()).concat(".html")));
      out.write("<html>\n<head>\n");
      out.write("</head>\n\n");
      out.write("<body bgcolor=\"white\">\n");
      while (enum.hasMoreElements()){
        txt=(String)enum.nextElement();
        out.write("<p>");
        out.write(txt);
        out.write("</p>");
        }
      out.write("</body>");
      out.write("</html>");
      out.close();
    } catch (IOException e) {
      System.out.println(" >> Fehler beim Schreiben");
      System.exit(1);
    }
  }
}

public void insert(String submenu, String txt){
  if (submenu != "" && hasSubMenus){
    int pos=getPos(submenu);
    if (pos==-1)
      vect.addElement(new Menu(submenu,"",txt));
    else
      ((Menu)vect.elementAt(pos)).insert(txt);
  }else if (submenu == "" && !hasSubMenus){
    this.insert(txt);
  }else {
    System.out.println(" >> Fehler: Inkorektes Vorlagenfile:");
    System.out.print("Zu ueberpruefen: Menustruktur, in welcher ");
    System.out.print("sich der Text \"");
    System.out.print(txt);
    System.out.println("\" befindet");
    System.exit(1);
  }
}

private void insert(String txt) {vect.addElement(txt);}

public String getName() {return name;}

public String getURL() {
  return name.toLowerCase().replace(' ','-').
  replace('ä','a').replace('ö','o').replace('ü','u');
}

public boolean hasSubMenus(){ return hasSubMenus;}

public int getNofSubmenus() {return (hasSubMenus) ? vect.size() : 0 ;}

public Vector getSubmenus() {return vect;}

public Menu getSubmenu(int i) {
  return (hasSubMenus) ? (Menu)vect.elementAt(i): null ;
}

private int getPos(String name){
      for (int i=0; i<vect.size(); i++)
    if ( ((Menu)vect.elementAt(i)).getName().equals(name) ) return i;
  return -1;
```

```
  }
}


class makepage{
    static public void main(String[] argv) {
    FileDialog fd;
    String file_in, file_out, dir_in, dir_out;
    Enumeration enum;

    System.out.println("\n===== Beginn Phase 1 (File reduzieren) =====");
      fd = new FileDialog(new Frame(), "Ausgangsdatei bestimmen", FileDialog.LOAD);
      fd.show();
      file_in=fd.getFile();
      dir_in=fd.getDirectory();
      if (dir_in==null){
        System.out.println("keine Datei eingegeben");
        System.exit(1);
      }
      fd = new FileDialog(new Frame(), "Zieldatei bestimmen", FileDialog.SAVE);
      fd.setDirectory(dir_in);
      fd.setFile("tmp_".concat(file_in));
      fd.show();
      file_out=fd.getFile();
      dir_out=fd.getDirectory();
      if (dir_in==null){
        System.out.println("keine Datei eingegeben");
        System.exit(1);
      }
      Reducer r = new Reducer(dir_in.concat(file_in),dir_out.concat(file_out));
      System.out.println("| oeffne Files                            |");
      if (!r.openFiles()) System.exit(1);
      System.out.println("| erstelle schlankes File                 |");
      if (!r.reduce()) System.exit(1);
     System.out.println("=========== Phase 1 erfolgreich ============");
    System.out.println("\n===== Beginn Phase 2 (Seite erstellen) =====");
      Vector groups =new Vector();
      System.out.println("| oeffne schlankes File                   |");
      Organizer o = new Organizer(dir_out, file_out);
      System.out.println("| erstelle Laufzeitstruktur               |");
      o.read();
      System.out.println("| Files erstellen (aus Vorlage)           |");
      o.write();

     System.out.println("============ Phase 2 erfolgreich ===========");
      enum=o.elements();
      System.out.println("");
      System.out.println("folgende Arbeitsgruppen wurden gefunden:");
      while (enum.hasMoreElements())
        System.out.println(((Group)enum.nextElement()).getName());

     System.exit(0);
  }
}
```

Anhang C

Pages

C.1 Willkommensseite

Diese Seite überprüft, ob im Browser Java, JavaScript und Cookies aktiviert sind. Ist dies der Fall, gelangt man via einen Link zur Hauptseite.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Willkommen zu MEDINO</title>

  <script type="text/javascript" language="JavaScript">
  <!--

  function CookieSetzen(name, wert, verfall){
    var jetzt = new Date();
    var Auszeit = new Date(jetzt.getTime() + verfall);
    document.cookie = name+"="+escape(wert)+"; expires="+Auszeit.toGMTString()+";";
  }

  function CookieLesen(name){
    var arg =name+"=";
    var alen=arg.length;
    var clen=document.cookie.length;
    var i=0;
    while (i<clen){
      var j =i +alen;
      if (document.cookie.substring(i,j)==arg) return getCookieVal(j);
      i=document.cookie.indexOf(" ",i)+1;
      if (i==0) break;
    }
    return null;
  }

    function getCookieVal (offset){
      var endstr=document.cookie.indexOf(";",offset);
      if (endstr==-1) endstr=document.cookie.length;
      return unescape(document.cookie.substring(offset,endstr));
   }
```

```
  // -->
  </script>

</head>

<body bgcolor="White" text="Black" link="Blue" vlink="Blue" alink="Blue">
<div align="center">
<h1>Willkommen zu MEDINO</h1>

<script type="text/javascript" language="JavaScript">
<!--

  var jahr = 1000*60*60*24*365;
  var stunde = 1000*60*60;
  var value;
  if (!document.cookie){ //falls diese Seite nie ein Cookie geschrieben hat:
    CookieSetzen('wissensintro', '1', jahr);
  }
  if (!document.cookie){ // falls nie Cookies geschrieben ==> also Cookies aus
    document.write("Bitte aktivieren Sie Cookies und laden Sie die Seite neu.<br>");
    document.write("(<a href=\"info/cookies.html\">Informationen zu Cookies</a>)<br><br>");
    if(!navigator.javaEnabled()){
      document.write("Bitte schalten Sie Java ein.<br>");
      document.write("(<a href=\"info/java.html\">Informationen zu Java</a>)<br><br>");
    }
  }else{
    value= CookieLesen('wissensintro');
    if (value==null)
      CookieSetzen('wissensintro', '1', jahr);
    CookieSetzen('hospital', 'insel', stunde);
    CookieSetzen('group','X',stunde);
    if(navigator.javaEnabled()){
      document.write("<a href=\"interviews/start.html\">");
      document.write("<img src=\"bilder/logo.jpg\" width=300 height=202 border=0>");
      document.write("</a>");
      document.write("<h1><a href=\"interviews/start.html\">START</a></h1>");
      document.write("(oder <a href=\"about/index.html\">Hintergrundinfos</a>)</p>");


    }
    document.write("<applet code=\"welcome.class\" codebase=\"java/\" width=1 height=1>");
    document.write("Bitte schalten Sie Java ein.<br>");
    document.write("(<a href=\"info/java.html\">Informationen zu Java</a>)<br><br>");
    document.write("</applet>");
  }

// -->
</script>

<noscript>
<applet code="welcome.class" codebase="java/" width=1 height=1>
Bitte schalten Sie Java ein.<br>
(<a href="info/java.html">Informationen zu Java</a>)<br><br>
</applet>

Bitte schalten Sie JavaScript ein.<br>
(<a href="info/js.html">Informationen zu JavaScript</a>)

</noscript>

</div>
```

```
</body>
</html>
```

C.2 Hauptseite

Hier geschieht die Wahl des Arbeitssystems oder des Wissenspunktes.

```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>MEDINO Hauptseite</title>

  <script type="text/javascript" language="JavaScript">
  <!--
  uszA = new Image(); uszA.src = "../buttons/usz.jpg";
  uszB = new Image(); uszB.src = "../buttons/usz1.jpg";
  leuvenA = new Image(); leuvenA.src = "../buttons/leuven.jpg";
  leuvenB = new Image(); leuvenB.src = "../buttons/leuven1.jpg";
  mayoA = new Image(); mayoA.src = "../buttons/mayo.jpg";
  mayoB = new Image(); mayoB.src = "../buttons/mayo1.jpg";
  inselA = new Image(); inselA.src = "../buttons/insel.jpg";
  inselB = new Image(); inselB.src = "../buttons/insel1.jpg";

  leer = new Image();
  leer.src = "help/leer.gif";
  about = new Image();
  about.src = "help/about.gif";
  wechselzeit_allgemein = new Image();
  wechselzeit_allgemein.src = "help/wechselzeit_allgemein.gif";
  wechselzeit_ausleitung = new Image();
  wechselzeit_ausleitung.src = "help/wechselzeit_ausleitung.gif";
  wechselzeit_lagerung = new Image();
  wechselzeit_lagerung.src = "help/wechselzeit_lagerung.gif";
  wechselzeit_mithilfe = new Image();
  wechselzeit_mithilfe.src = "help/wechselzeit_mithilfe.gif";
  wechselzeit_operateur = new Image();
  wechselzeit_operateur.src = "help/wechselzeit_operateur.gif";
  holdingarea_intimsphare = new Image();
  holdingarea_intimsphare.src = "help/holdingarea_intimsphare.gif";
  holdingarea_organisation = new Image();
  holdingarea_organisation.src = "help/holdingarea_organisation.gif";
  aufwachraum = new Image();
  aufwachraum.src = "help/aufwachraum.gif";
  grossraumop = new Image();
  grossraumop.src = "help/grossraumop.gif";
  teilbarerop = new Image();
  teilbarerop.src = "help/teilbarerop.gif";
  kombinationab = new Image();
  kombinationab.src = "help/kombinationab.gif";
  idee = new Image();
  idee.src = "help/idee.gif";
  information = new Image();
  information.src = "help/information.gif";
  kommentar_allgemein = new Image();
  kommentar_allgemein.src = "help/kommentar_allgemein.gif";
  kommentar_bettenbewirtschaftung = new Image();
  kommentar_bettenbewirtschaftung.src = "help/kommentar_bettenbewirtschaftung.gif";
  kommentar_intensivstation = new Image();
```

```
kommentar_intensivstation.src = "help/kommentar_intensivstation.gif";
kommentar_konzeptfrage = new Image();
kommentar_konzeptfrage.src = "help/kommentar_konzeptfrage.gif";
kommentar_lagerung = new Image();
kommentar_lagerung.src = "help/kommentar_lagerung.gif";
kommentar_studie = new Image();
kommentar_studie.src = "help/kommentar_studie.gif";
konzepta_anzahlraume = new Image();
konzepta_anzahlraume.src = "help/konzepta_anzahlraume.gif";
konzepta_ausbildung = new Image();
konzepta_ausbildung.src = "help/konzepta_ausbildung.gif";
konzepta_gerate = new Image();
konzepta_gerate.src = "help/konzepta_gerate.gif";
konzepta_kleineeingriffe = new Image();
konzepta_kleineeingriffe.src = "help/konzepta_kleineeingriffe.gif";
konzepta_narkosebelastung = new Image();
konzepta_narkosebelastung.src = "help/konzepta_narkosebelastung.gif";
konzepta_patientenbetreuung = new Image();
konzepta_patientenbetreuung.src = "help/konzepta_patientenbetreuung.gif";
konzepta_patientenkomfort = new Image();
konzepta_patientenkomfort.src = "help/konzepta_patientenkomfort.gif";
konzepta_patiententransport = new Image();
konzepta_patiententransport.src = "help/konzepta_patiententransport.gif";
konzepta_personalbedarf = new Image();
konzepta_personalbedarf.src = "help/konzepta_personalbedarf.gif";
konzepta_saalvorbereitung = new Image();
konzepta_saalvorbereitung.src = "help/konzepta_saalvorbereitung.gif";
konzepta_teamperformance = new Image();
konzepta_teamperformance.src = "help/konzepta_teamperformance.gif";
konzepta_wechselzeit = new Image();
konzepta_wechselzeit.src = "help/konzepta_wechselzeit.gif";
konzeptb_anzahlraume = new Image();
konzeptb_anzahlraume.src = "help/konzeptb_anzahlraume.gif";
konzeptb_ausbildung = new Image();
konzeptb_ausbildung.src = "help/konzeptb_ausbildung.gif";
konzeptb_gerate = new Image();
konzeptb_gerate.src = "help/konzeptb_gerate.gif";
konzeptb_hygiene = new Image();
konzeptb_hygiene.src = "help/konzeptb_hygiene.gif";
konzeptb_material = new Image();
konzeptb_material.src = "help/konzeptb_material.gif";
konzeptb_narkosebelastung = new Image();
konzeptb_narkosebelastung.src = "help/konzeptb_narkosebelastung.gif";
konzeptb_organisation = new Image();
konzeptb_organisation.src = "help/konzeptb_organisation.gif";
konzeptb_patientenbetreuung = new Image();
konzeptb_patientenbetreuung.src = "help/konzeptb_patientenbetreuung.gif";
konzeptb_patientenkomfort = new Image();
konzeptb_patientenkomfort.src = "help/konzeptb_patientenkomfort.gif";
konzeptb_patiententransport = new Image();
konzeptb_patiententransport.src = "help/konzeptb_patiententransport.gif";
konzeptb_personalbedarf = new Image();
konzeptb_personalbedarf.src = "help/konzeptb_personalbedarf.gif";
konzeptb_saalvorbereitung = new Image();
konzeptb_saalvorbereitung.src = "help/konzeptb_saalvorbereitung.gif";
konzeptb_teamperformance = new Image();
konzeptb_teamperformance.src = "help/konzeptb_teamperformance.gif";
konzeptb_wechselzeit = new Image();
konzeptb_wechselzeit.src = "help/konzeptb_wechselzeit.gif";
koordination = new Image();
koordination.src = "help/koordination.gif";
material = new Image();
```

```
material.src = "help/material.gif";
multivorbereitung = new Image();
multivorbereitung.src = "help/multivorbereitung.gif";
personal = new Image();
personal.src = "help/personal.gif";
programm_anmeldeschluss = new Image();
programm_anmeldeschluss.src = "help/programm_anmeldeschluss.gif";
programm_kriterien = new Image();
programm_kriterien.src = "help/programm_kriterien.gif";
programm_mitgestaltung = new Image();
programm_mitgestaltung.src = "help/programm_mitgestaltung.gif";
programm_planungshorizont = new Image();
programm_planungshorizont.src = "help/programm_planungshorizont.gif";
programm_praoperativeablaufe = new Image();
programm_praoperativeablaufe.src = "help/programm_praoperativeablaufe.gif";
programm_startzeit = new Image();
programm_startzeit.src = "help/programm_startzeit.gif";
programm_teamplanung = new Image();
programm_teamplanung.src = "help/programm_teamplanung.gif";
programm_verbindlichkeit = new Image();
programm_verbindlichkeit.src = "help/programm_verbindlichkeit.gif";
saalauslastung = new Image();
saalauslastung.src = "help/saalauslastung.gif";
saalzuteilung = new Image();
saalzuteilung.src = "help/saalzuteilung.gif";
vergleichsspitaler = new Image();
vergleichsspitaler.src = "help/vergleichsspitaler.gif";
zentralervorbereitungsraum = new Image();
zentralervorbereitungsraum.src = "help/zentralervorbereitungsraum.gif";
optimaleropbetrieb = new Image();
optimaleropbetrieb.src = "help/optimaleropbetrieb.gif";
ausbildung = new Image();
ausbildung.src = "help/ausbildung.gif";

function setandgo(subject){
  CookieSetzen('subject', subject, 1000*60*60);
  CookieSetzen('submenus', '0', 1000*60*60);
  top.location.href= "index1.html";
}

function CookieSetzen(name, wert, verfall){
  var jetzt = new Date();
  var Auszeit = new Date(jetzt.getTime() + verfall);
  document.cookie = name+"="+escape(wert)+"; expires="+Auszeit.toGMTString()+";";
}

function CookieLesen(name){
  var arg =name+"=";
  var alen=arg.length;
  var clen=document.cookie.length;
  var i=0;
  while (i<clen){
    var j =i +alen;
    if (document.cookie.substring(i,j)==arg) return getCookieVal(j);
    i=document.cookie.indexOf(" ",i)+1;
    if (i==0) break;
  }
  return null;
}

  function getCookieVal (offset){
    var endstr=document.cookie.indexOf(";",offset);
```

```
      if (endstr==-1) endstr=document.cookie.length;
      return unescape(document.cookie.substring(offset,endstr));
   }

  //-->
  </script>

</head>

<body background="../background/orange.gif" bgcolor="White"
  text="Black" link="Blue" vlink="Blue" alink="Blue">

<table border="0" cellspacing="2" cellpadding="2">
<tr>
<th><h2>Arbeitssystem</h2></th>
<th><h2>Wissensthemen</h2></th>
</tr>
<tr>
<td> <!-- Spalte mit Spitalauswahl -->
  <table border="0" cellspacing="2" cellpadding="2">
  <tr>
    <td align="CENTER" valign="TOP">
      <a href="../architektur/insel.html" onMouseOver="window.document.insel.src=inselB.src"
      onMouseOut="window.document.insel.src=inselA.src">
      <img name="insel" src="../buttons/insel.jpg" width=190
        height=100 border=0 alt="Inselspital Bern" ><br>
      <font size="-1"><b>Inselspital Bern</b></font>
      </a>
    </td>
  </tr>
  <tr>
    <td align="CENTER" valign="TOP">
      <a href="../architektur/usz.html" onMouseOver="window.document.usz.src=uszB.src"
        onMouseOut="window.document.usz.src=uszA.src">
        <img name="usz" src="../buttons/usz.jpg" width=190 height=100 border=0
          alt="Universit&auml;tsspital Z&uuml;rich" ><br>
        <font size="-1"><b>Universit&auml;tsspital Z&uuml;rich</b></font>
      </a>
    </td>
  </tr>
  <tr>
    <td align="CENTER" valign="TOP">
      <a href="../architektur/leuven.html" onMouseOver="window.document.leuven.src=leuvenB.src"
        onMouseOut="window.document.leuven.src=leuvenA.src">
        <img name="leuven" src="../buttons/leuven.jpg" width=190 height=100 border=0
          alt="Universit&auml;tsspital Leuven/Belgien"><br>
        <font size="-1"><b>Universit&auml;tsspital Leuven/Belgien</b></font>
      </a>
    </td>
  </tr>
  <tr>
    <td align="CENTER" valign="TOP">
      <a href="../architektur/mayo.html" onMouseOver="window.document.mayo.src=mayoB.src"
        onMouseOut="window.document.mayo.src=mayoA.src">
        <img name="mayo" src="../buttons/mayo.jpg" width=190 height=100 border=0
          alt="Mayo Medical Center in Rochster/USA"><br>
        <font size="-1"><b>Mayo Medical Center in Rochster/USA</b></font>
      </a>
      </td>
  </tr>
</table>
</td>
```

```
<td align="CENTER" valign="TOP"> <!-- Spalte mit Wissendaten -->
  <center>
  <img name= "mindmap" src="mindmap.gif" border="0" usemap="#mindmap">
  <table border=1>
  <tr>
    <td>
      <img name="help" src="help/leer.gif" width=500 height=35 border=0>
    </td>
  </tr>
  </table>
  </center>
<map name="mindmap">
<area shape=polygon coords="192,155,230,135,265,155,230,175"
  href="../about/index.html"
  onmouseover="window.document.help.src=about.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="96,7,146,23"
  href="javascript:setandgo('wechselzeit_allgemein')"
  onmouseover="window.document.help.src=wechselzeit_allgemein.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="92,22,146,38"
  href="javascript:setandgo('wechselzeit_ausleitung')"
  onmouseover="window.document.help.src=wechselzeit_ausleitung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="97,37,146,53"
  href="javascript:setandgo('wechselzeit_lagerung')"
  onmouseover="window.document.help.src=wechselzeit_lagerung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="105,52,146,69"
  href="javascript:setandgo('wechselzeit_mithilfe')"
  onmouseover="window.document.help.src=wechselzeit_mithilfe.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="94,67,146,84"
  href="javascript:setandgo('wechselzeit_operateur')"
  onmouseover="window.document.help.src=wechselzeit_operateur.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="347,86,407,103"
  href="javascript:setandgo('holding-area_intimsphare')"
  onmouseover="window.document.help.src=holdingarea_intimsphare.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="347,102,410,118"
  href="javascript:setandgo('holding-area_organisation')"
  onmouseover="window.document.help.src=holdingarea_organisation.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="250,37,331,55"
  href="javascript:setandgo('aufwachraum')"
  onmouseover="window.document.help.src=aufwachraum.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="265,68,348,86"
  href="javascript:setandgo('grossraum-op')"
  onmouseover="window.document.help.src=grossraumop.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="144,71,214,89"
  href="javascript:setandgo('teilbarer-op')"
  onmouseover="window.document.help.src=teilbarerop.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="244,197,337,215"
  href="javascript:setandgo('kombination-a+b')"
  onmouseover="window.document.help.src=kombinationab.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="264,113,295,131"
  href="javascript:setandgo('idee')"
```

```
  onmouseover="window.document.help.src=idee.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="271,131,335,149"
  href="javascript:setandgo('information')"
  onmouseover="window.document.help.src=information.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,122,395,139"
  href="javascript:setandgo('kommentar_allgemein')"
  onmouseover="window.document.help.src=kommentar_allgemein.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,137,449,154"
  href="javascript:setandgo('kommentar_bettenbewirtschaftung')"
  onmouseover="window.document.help.src=kommentar_bettenbewirtschaftung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,152,418,169"
  href="javascript:setandgo('kommentar_intensivstation')"
  onmouseover="window.document.help.src=kommentar_intensivstation.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,168,411,184"
  href="javascript:setandgo('kommentar_konzeptfrage')"
  onmouseover="window.document.help.src=kommentar_konzeptfrage.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,183,394,199"
  href="javascript:setandgo('kommentar_lagerung')"
  onmouseover="window.document.help.src=kommentar_lagerung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="345,198,382,214"
  href="javascript:setandgo('kommentar_studie')"
  onmouseover="window.document.help.src=kommentar_studie.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,217,370,233"
  href="javascript:setandgo('konzept-a_anzahl-raume')"
  onmouseover="window.document.help.src=konzepta_anzahlraume.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,232,355,248"
  href="javascript:setandgo('konzept-a_ausbildung')"
  onmouseover="window.document.help.src=konzepta_ausbildung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,247,338,263"
  href="javascript:setandgo('konzept-a_gerate')"
  onmouseover="window.document.help.src=konzepta_gerate.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,262,376,279"
  href="javascript:setandgo('konzept-a_kleine-eingriffe')"
  onmouseover="window.document.help.src=konzepta_kleineeingriffe.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,277,385,294"
  href="javascript:setandgo('konzept-a_narkosebelastung')"
  onmouseover="window.document.help.src=konzepta_narkosebelastung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,292,390,309"
  href="javascript:setandgo('konzept-a_patientenbetreuung')"
  onmouseover="window.document.help.src=konzepta_patientenbetreuung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,308,382,324"
  href="javascript:setandgo('konzept-a_patientenkomfort')"
  onmouseover="window.document.help.src=konzepta_patientenkomfort.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,323,386,339"
  href="javascript:setandgo('konzept-a_patiententransport')"
  onmouseover="window.document.help.src=konzepta_patiententransport.src"
  onmouseout="window.document.help.src=leer.src">
```

```
<area shape="rect" coords="299,338,373,354"
  href="javascript:setandgo('konzept-a_personalbedarf')"
  onmouseover="window.document.help.src=konzepta_personalbedarf.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,353,380,370"
  href="javascript:setandgo('konzept-a_saalvorbereitung')"
  onmouseover="window.document.help.src=konzepta_saalvorbereitung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,368,386,385"
  href="javascript:setandgo('konzept-a_teamperformance')"
  onmouseover="window.document.help.src=konzepta_teamperformance.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="299,383,360,400"
  href="javascript:setandgo('konzept-a_wechselzeit')"
  onmouseover="window.document.help.src=konzepta_wechselzeit.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="65,201,136,218"
  href="javascript:setandgo('konzept-b_anzahl-raume')"
  onmouseover="window.document.help.src=konzeptb_anzahlraume.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="80,217,136,233"
  href="javascript:setandgo('konzept-b_ausbildung')"
  onmouseover="window.document.help.src=konzeptb_ausbildung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="97,232,136,248"
  href="javascript:setandgo('konzept-b_gerate')"
  onmouseover="window.document.help.src=konzeptb_gerate.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="90,247,136,263"
  href="javascript:setandgo('konzept-b_hygiene')"
  onmouseover="window.document.help.src=konzeptb_hygiene.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="90,262,136,279"
  href="javascript:setandgo('konzept-b_material')"
  onmouseover="window.document.help.src=konzeptb_material.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="50,277,136,294"
  href="javascript:setandgo('konzept-b_narkosebelastung')"
  onmouseover="window.document.help.src=konzeptb_narkosebelastung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="73,292,136,309"
  href="javascript:setandgo('konzept-b_organisation')"
  onmouseover="window.document.help.src=konzeptb_organisation.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="45,308,136,324"
  href="javascript:setandgo('konzept-b_patientenbetreuung')"
  onmouseover="window.document.help.src=konzeptb_patientenbetreuung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="53,323,136,339"
  href="javascript:setandgo('konzept-b_patientenkomfort')"
  onmouseover="window.document.help.src=konzeptb_patientenkomfort.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="49,338,136,354"
  href="javascript:setandgo('konzept-b_patiententransport')"
  onmouseover="window.document.help.src=konzeptb_patiententransport.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="62,353,136,370"
  href="javascript:setandgo('konzept-b_personalbedarf')"
  onmouseover="window.document.help.src=konzeptb_personalbedarf.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="55,368,136,385"
  href="javascript:setandgo('konzept-b_saalvorbereitung')"
```

```
  onmouseover="window.document.help.src=konzeptb_saalvorbereitung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="49,383,136,400"
  href="javascript:setandgo('konzept-b_teamperformance')"
  onmouseover="window.document.help.src=konzeptb_teamperformance.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="75,399,136,415"
  href="javascript:setandgo('konzept-b_wechselzeit')"
  onmouseover="window.document.help.src=konzeptb_wechselzeit.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="147,238,218,256"
  href="javascript:setandgo('koordination')"
  onmouseover="window.document.help.src=koordination.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="155,212,203,230"
  href="javascript:setandgo('material')"
  onmouseover="window.document.help.src=material.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="120,186,219,204"
  href="javascript:setandgo('multi-vorbereitung')"
  onmouseover="window.document.help.src=multivorbereitung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="124,148,177,166"
  href="javascript:setandgo('personal')"
  onmouseover="window.document.help.src=personal.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="25,78,104,94"
  href="javascript:setandgo('programm_anmeldeschluss')"
  onmouseover="window.document.help.src=programm_anmeldeschluss.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="59,93,104,110"
  href="javascript:setandgo('programm_kriterien')"
  onmouseover="window.document.help.src=programm_kriterien.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="38,108,104,125"
  href="javascript:setandgo('programm_mitgestaltung')"
  onmouseover="window.document.help.src=programm_mitgestaltung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="23,123,104,140"
  href="javascript:setandgo('programm_planungshorizont')"
  onmouseover="window.document.help.src=programm_planungshorizont.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="5,139,104,155"
  href="javascript:setandgo('programm_praoperative-ablaufe')"
  onmouseover="window.document.help.src=programm_praoperativeablaufe.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="59,154,104,170"
  href="javascript:setandgo('programm_startzeit')"
  onmouseover="window.document.help.src=programm_startzeit.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="38,169,104,185"
  href="javascript:setandgo('programm_teamplanung')"
  onmouseover="window.document.help.src=programm_teamplanung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="33,184,104,201"
  href="javascript:setandgo('programm_verbindlichkeit')"
  onmouseover="window.document.help.src=programm_verbindlichkeit.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="118,113,203,131"
  href="javascript:setandgo('saalauslastung')"
  onmouseover="window.document.help.src=saalauslastung.src"
  onmouseout="window.document.help.src=leer.src">
```

```
<area shape="rect" coords="127,101,202,119"
  href="javascript:setandgo('saalzuteilung')"
  onmouseover="window.document.help.src=saalzuteilung.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="118,87,216,105"
  href="javascript:setandgo('vergleichsspitaler')"
  onmouseover="window.document.help.src=vergleichsspitaler.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="233,12,384,30"
  href="javascript:setandgo('zentraler-vorbereitungsraum')"
  onmouseover="window.document.help.src=zentralervorbereitungsraum.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="108,170,217,188"
  href="javascript:setandgo('optimaler-op-betrieb')"
  onmouseover="window.document.help.src=optimaleropbetrieb.src"
  onmouseout="window.document.help.src=leer.src">
<area shape="rect" coords="258,50,323,68"
  href="javascript:setandgo('ausbildung')"
  onmouseover="window.document.help.src=ausbildung.src"
  onmouseout="window.document.help.src=leer.src">
</map>
</td>
</tr>
</table>

</body>
</html>
```

C.3 Architektur

am Beispiel des Insel-Spitals

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Inselspital Bern: Architektur</title>
</head>

<body bgcolor="White" text="Black" link="Blue" vlink="Blue" alink="Blue">

<table border="0" cellspacing="0" cellpadding="2" bgcolor="FFBB00">
<tr>
    <td width="100%-190"align="CENTER" valign="MIDDLE">
      <h1>Inselspital Bern</h1>
      <a href="../interviews/start.html">Hauptseite</a> |
      Architektur |
      <a href="../ablauf/insel.html">Arbeitsablauf</a>
    </td>
    <td width="190">
    <a href="../interviews/start.html">
      <img src="../buttons/insel1.jpg" width=190 height=100 border=0 alt="Inselspital Bern">
    </a>
    </td>
</tr>
</table>

<p> <br>
</p>
```

```
<center>
<applet code="intro.class" codebase="../java/" width=600 height=330>
<param name="spital" value="insel">
<param name="standard" value="Wenn sich die Maus über einem der Bilder befindet,
erscheint hier die Beschreibung des Bildbereichs.">

<param name="gr1" value="50|40|106|88|OP6: Gefässchirurgie">
<param name="gr2" value="130|40|180|88|OP5: Gefässchirurgie">
<param name="gr3" value="225|40|278|88|OP4: Plastische und Wiederherstellungschirurgie">
<param name="gr4" value="298|40|352|88|OP3: Gefässchirurgie">
<param name="gr5" value="400|40|450|88|OP2: Thoraxchirurgie">
<param name="gr6" value="472|40|525|88|OP1: Herzchirurgie">

<param name="gr7" value="50|88|87|121|Vorbereitungsraum">
<param name="gr8" value="147|88|180|121|Vorbereitungsraum">
<param name="gr9" value="225|88|262|121|Vorbereitungsraum">
<param name="gr10" value="319|88|353|121|Vorbereitungsraum">
<param name="gr11" value="400|88|435|121|Vorbereitungsraum">
<param name="gr12" value="489|88|524|121|Vorbereitungsraum">

<param name="gr13" value="87|85|147|121|Ausleitungsraum">
<param name="gr14" value="262|85|319|121|Ausleitungsraum">
<param name="gr15" value="435|85|489|121|Ausleitungsraum">


<param name="gr16" value="106|40|130|85|Instrumentenaufbereitung">
<param name="gr17" value="278|40|298|85|Instrumentenaufbereitung">
<param name="gr18" value="450|40|472|85|Instrumentenaufbereitung">

<param name="gr19" value="536|123|556|136|Patienten-Eingang">
<param name="gr20" value="63|121|536|140|Patientengang">


<param name="det1" value="6|3|97|93|OP4">
<param name="det2" value="145|3|237|93|OP3">

<param name="det3" value="4|93|68|155|Vorbereitungsraum">
<param name="det4" value="68|93|175|155|Ausleitungsraum">
<param name="det5" value="175|93|237|155|Vorbereitungsraum">

<param name="det6" value="100|3|142|82|Instrumentenaufbereitung">

</applet>
</center>

<blockquote>
<p>
<B>Daten 1997:</B> Betten: 1060, station&auml;re Patienten: 36745, Durchschnittliche
Aufenthaltsdauer: 8,8 Tage<BR>
Der OP-Trakt E-West des Inselspital Bern hat 6 OP-S&auml;le und wurde
1964 in Betrieb genommen.
</p>
<p>
<b>Details:</b>
In den Wand- und T&uuml;renelementen zwischen den OP-S&auml;len und dem Sterilgang sind
nachtr&auml;glich grossfl&auml;chige Fenster eingebaut worden.
Die Raumteilung zwischen OP-Saal und Vorbereitungsraum ist jedoch im urspr&uuml;nglichen
Zustand, mit nur seht kleinen T&uuml;rfenstern, belassen worden.
<br>
Alle Versorgungsanschl&uuml;sse (Energie, Gase) sind in den Vorbereitungs- und
Ausleitungsr&auml;umen, sowie in den OP-S&auml;len als Wandanschl&uuml;sse ausgef&uuml;hrt.
```

```
Die notwendigen Verbindungsleitungen (Kabel, Schl&auml;uche) stellen eine Gef&auml;hrdung
(Stolpergefahr, Materialverschleiss) dar.
</p>

</blockquote>

</body>
</html>
```

C.4 Arbeitsabläufe

am Beispiel des Insel-Spitals

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Inselspital Bern: Arbeitsabl&auml;ufe</title>

  <script type="text/javascript" language="JavaScript">
    function go(){
      F1 = open("legende.html","Legende","width=650,height=250,resizable=yes");
    }
  </script>

</head>


<body bgcolor="White" text="Black" link="Blue" vlink="Blue" alink="Blue">

<table border="0" cellspacing="0" cellpadding="2" bgcolor="FFBB00">
<tr>
  <td width="100%-190"align="CENTER" valign="MIDDLE">
    <h1>Inselspital Bern</h1>
    <a href="../interviews/start.html">Hauptseite</a> |
    <a href="../architektur/insel.html">Architektur</a> |
    Arbeitsablauf
  </td>
  <td width="190">
    <a href="../interviews/start.html">
    <img src="../buttons/insel1.jpg" width=190 height=100 border=0
      alt="Inselspital Bern">
</a>
  </td>
</tr>
</table>


<center>
<p> <br>
<applet code="ablauf.class" codebase="../java/" width=620 height=300>
<param name="spital" value="insel">
<param name="anzahl_bilder" value="8">
<param name="format_ablaufbilder" value=".gif">
<param name="text1" value="Dies ist der erste Text zum ersten Bild.">
<param name="text2" value="Dies ist der 2. Text.">
<param name="text3" value="Dies ist der 3. Text.">
<param name="text4" value="Dies ist der 4. Text.">
<param name="text5" value="Dies ist der 5. Text.">
```

```
<param name="text6" value="Dies ist der 6. Text.">
<param name="text7" value="Dies ist der 7. Text.">
<param name="text8" value="Dies ist der 8. Text.">
</applet>
<a href="javascript:go()">Legende</a></p>
</center>
</body>
</html>
```

C.5 Wissensthemen

C.5.1 Vergleichsmodus

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Wissensthemen</title>

  <script type="text/javascript" language="JavaScript">
  <!--

  function setandgo(url){
    var jahr = 1000*60*60*24*365;
    var stunde = 1000*60*60;
    var slash = "\\";
    var path = top.Ltext.location.pathname;
    var end= "";
    var dir = "";
    var paths = "";
    var datei = "";
    var hosp = "";
    var group ="";

    end = path.lastIndexOf(slash);
    if (end == -1) {
      slash="/";
      end =path.lastIndexOf(slash);
    }

    dir = path.slice(0,end);
    paths = path.split(slash);
    datei = paths[paths.length-1];
    hosp = paths[paths.length-2];
    if (datei == "leer.html" || datei == "Lintro.html" || datei == "mindmap.html"){
      path = top.Lmenu.location.pathname;
      paths = path.split(slash);
      datei = paths[paths.length-1];
      hosp = paths[paths.length-2];
      end = datei.indexOf("_");
      if (end == -1)
        end = datei.indexOf(".");
      group=datei.slice(2,end); // Lm resp. Rm ignorieren
    }else{
      end = datei.indexOf("_");
      group=datei.slice(0,end);
    }
    CookieSetzen('hospital', hosp, stunde);
```

```
    CookieSetzen('group', group, stunde);
    top.location.href= url;

  }

  function openhelp(val){
    if (val =="1")
      HELP = open("help.html","Hilfe","width=600,height=450,resizable=yes");
  }

  function CookieSetzen(name, wert, verfall){
    var jetzt = new Date();
    var Auszeit = new Date(jetzt.getTime() + verfall);
    document.cookie = name+"="+escape(wert)+"; expires="+Auszeit.toGMTString()+";";
  }

  function CookieLesen(name){
    var arg =name+"=";
    var alen=arg.length;
    var clen=document.cookie.length;
    var i=0;
    while (i<clen){
      var j =i +alen;
      if (document.cookie.substring(i,j)==arg) return getCookieVal(j);
      i=document.cookie.indexOf(" ",i)+1;
      if (i==0) break;
    }
    return null;
  }

  function getCookieVal (offset){
    var endstr=document.cookie.indexOf(";",offset);
    if (endstr==-1) endstr=document.cookie.length;
    return unescape(document.cookie.substring(offset,endstr));
  }

  // -->
  </script>

</head>


<!-- frames -->
  <script type="text/javascript" language="JavaScript">
  <!--
  var hospital= CookieLesen('hospital');
  var group= CookieLesen('group');
  var subject= CookieLesen('subject');
  var submenus= CookieLesen('submenus');
//  var help= CookieLesen('wissensintro');
  var slash = "\\";
  var path = top.location.pathname;
  var paths = path.split(slash);
  var urlt;
  var urlmL;
  var urlmR;
  if (paths.length == 1){
    slash = "/"
    paths = path.split(slash);
  }
  urlt=hospital+slash+group+"_"+subject+".html";
  ulrmL=hospital+slash+"Lm"+group+"_"+subject+".html#this";
```

```
  ulrmR=hospital+slash+"Rm"+group+"_"+subject+".html#this";
  if (submenus == "1") urlt="leer.html";
//  openhelp(help);


document.write("<frameset  rows=\"120,*\">");
  document.write("<frame name=\"head\" src=\"head2.html\" marginwidth=\"10\"");
  document.write("marginheight=\"10\" scrolling=\"no\" frameborder=\"no\">");
  document.write("<frameset  cols=\"25%,*,25%\">");
    document.write("<frameset  rows=\"100,*\">");
      document.write("<frame name=\"Lselect\" src=\"Lselect.html\" marginwidth=\"10\"");
      document.write("marginheight=\"10\" scrolling=\"no\" frameborder=\"no\">");
       document.write("<frame name=\"Lmenu\" src=\"");document.write(ulrmL);
       document.write("\" marginwidth=\"10\" marginheight=\"10\" scrolling=\"auto\"");
       document.write("frameborder=\"no\">");
    document.write("</frameset>");
    document.write("<frameset  rows=\"50%,*\">");
      document.write("<frame name=\"Ltext\" src=\"");document.write(urlt);
      document.write("\" marginwidth=\"10\" marginheight=\"10\" scrolling=\"auto\"");
      document.write("frameborder=\"yes\">");
      document.write("<frame name=\"Rtext\" src=\"");document.write(urlt);
      document.write("\" marginwidth=\"10\" marginheight=\"10\" scrolling=\"auto\"");
      document.write("frameborder=\"yes\">");
    document.write("</frameset>");
    document.write("<frameset  rows=\"100,*\">");
      document.write("<frame name=\"Rselect\" src=\"Rselect.html\" marginwidth=\"10\"");
      document.write("marginheight=\"10\" scrolling=\"no\" frameborder=\"no\">");
      document.write("<frame name=\"Rmenu\" src=\"");document.write(ulrmR);
      document.write("\" marginwidth=\"10\" marginheight=\"10\" scrolling=\"auto\"");
      document.write("frameborder=\"no\">");
    document.write("</frameset>");
  document.write("</frameset>");
document.write("</frameset>");

  // -->
  </script>

<!-- frames -->




<body>
</body>
</html>
```

C.5.2 Gesammt-Navigation

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <script type="text/javascript" language="JavaScript">
  <!--

  function setandgo(url){
    var jahr = 1000*60*60*24*365;
    var stunde = 1000*60*60;
    var slash = "\\";
    var path = top.Ltext.location.pathname;
```

```
    var end= "";
    var start= "";
    var dir = "";
    var paths = "";
    var datei = "";
    var hosp = "";
    var group ="";
    var smenu ="";
var submenus ="0";

    end = path.lastIndexOf(slash);
    if (end == -1) {
      slash="/";
      end =path.lastIndexOf(slash);
    }

    dir = path.slice(0,end);
    paths = path.split(slash);
    datei = paths[paths.length-1];
    hosp = paths[paths.length-2];
    if (datei == "leer.html"){ // man befindet sich in einem Haupast, mit Unteraesten
      submenus ="1";
      path = top.Lmenu.location.pathname;
      paths = path.split(slash);
      datei = paths[paths.length-1];
      hosp = paths[paths.length-2];
      start = datei.indexOf("_");
      end = datei.indexOf(".");
      group=datei.slice(2,start); // Lm resp. Rm ignorieren
      menu = datei.slice(start+1,end);
      smenu ="";
    }else{
      start=datei.indexOf("_");
      end=datei.lastIndexOf("_");
      if (end==start){ // keine submenus
        end=datei.indexOf(".");
        menu = datei.slice(start+1,end);
        smenu= "";
      }else{
        menu = datei.slice(start+1,end)+"_";
        smenu = datei.slice(end+1,datei.indexOf("."));
      }
      end = datei.indexOf("_");
      group=datei.slice(0,end);
    }
    CookieSetzen('hospital', hosp, stunde);
    CookieSetzen('group', group, stunde);
    CookieSetzen('subject', menu+smenu, stunde);
    CookieSetzen('submenus', submenus, stunde);
    top.location.href= url;
  }

  function CookieSetzen(name, wert, verfall){
    var jetzt = new Date();
    var Auszeit = new Date(jetzt.getTime() + verfall);
    document.cookie = name+"="+escape(wert)+"; expires="+Auszeit.toGMTString()+";";
  }

  function CookieLesen(name){
    var arg =name+"=";
    var alen=arg.length;
    var clen=document.cookie.length;
```

```
    var i=0;
    while (i<clen){
      var j =i +alen;
      if (document.cookie.substring(i,j)==arg) return getCookieVal(j);
      i=document.cookie.indexOf(" ",i)+1;
      if (i==0) break;
    }
    return null;
  }

  function getCookieVal (offset){
    var endstr=document.cookie.indexOf(";",offset);
    if (endstr==-1) endstr=document.cookie.length;
    return unescape(document.cookie.substring(offset,endstr));
  }

  function openhelp(){
    HELP = open("help1.html","Hilfe","width=600,height=350,resizable=yes");
  }

  // -->
  </script>

</head>

<body bgcolor="#FFBB00" text="Black" link="Blue" vlink="Blue" alink="Blue">

<table border="0" cellspacing="0" cellpadding="2">
<tr>
    <td width="100%-290"align="CENTER" valign="MIDDLE">
    <h1>Wissensthemen</h1>
    <a href="javascript:setandgo('start.html')">Hauptseite</a> |
    <a href="javascript:setandgo('index2.html')">Vergleich</a> |
    <a href="javascript:setandgo('feedback.html')">Kommentar</a> |
    <a href="javascript:openhelp()">Navigationshilfe</a>
    </td>
    <td valign="TOP">
    <a href="javascript:setandgo('start.html')">
    <img src="mindmap_little.gif" width=171 height=90 border=0 alt="">
    </a>
  </td>
</tr>
</table>

</body>
</html>
```

C.5.3 Spital und Berufsgruppenwahl

am Bespiel des linken Frames

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>leer</title>

  <script language="JavaScript">
  <!--
```

```
function changeH(hosp){
  var slash = "\\";
  var path = top.Ltext.location.pathname;
  var paths;
  var datei;
  var dateis;
  var group;
  var mURL;

  paths = path.split(slash);
  if (paths.length == 1){
    slash = "/"
    paths = path.split(slash);
  }
  datei = paths[paths.length-1];
  dateis = datei.split("_");
  group = "Lm"+dateis[0];
  mURL = "";

  if (datei == "leer.html" || datei == "Lintro.html" || datei == "mindmap.html"){
    path = top.Lmenu.location.pathname;
    paths = path.split(slash);
    datei = paths[paths.length-1];
    top.Lmenu.location.href=hosp+slash+datei;
  }else{
    top.Ltext.location.href=hosp+slash+datei;
    mURL=group;
    if (dateis.length==1)
      mURL+=".html";
    for (i=1; i<dateis.length; i++)
      mURL+="_"+dateis[i];
    top.Lmenu.location.href=hosp+slash+mURL+"#this";
  }
}

function changeG(group){
  var slash = "\\";
  var path = top.Ltext.location.pathname;
  var end;
  var dir;
  var paths;
  var datei;
  var hosp;
  var start;
  var menu;
  var smenu;

  end = path.lastIndexOf(slash);
  if (end == -1) {
    slash="/";
    end =path.lastIndexOf(slash);
  }

  dir = path.slice(0,end);
  paths = path.split(slash);
  datei = paths[paths.length-1];
  hosp = paths[paths.length-2];;
  if (datei == "leer.html" || datei == "Lintro.html" || datei == "mindmap.html"){
    path = top.Lmenu.location.pathname;
    paths = path.split(slash);
    datei = paths[paths.length-1];
```

```
      hosp = paths[paths.length-2];
      start=datei.indexOf("_");
    if (start == -1){
      menu="";
      }else{
        end=datei.lastIndexOf(".");
        menu = datei.slice(start,end);
      }
      top.Lmenu.location.href=hosp+slash+"Lm"+group+menu+".html#this";
    }else{
      //menu filtern _menu_ oder _menu.
      start=datei.indexOf("_");
      end=datei.lastIndexOf("_");
      if (end==start){ // keine submenus
        end=datei.indexOf(".");
        menu = datei.slice(start,end);
        smenu= "";
      }else{
        menu = datei.slice(start,end)+"_";
        smenu = datei.slice(end+1,datei.indexOf("."));
      }
      top.Ltext.location.href=hosp+slash+group+menu+smenu+".html";
      top.Lmenu.location.href=hosp+slash+"Lm"+group+menu+smenu+".html#this";
    }
  }

  function CookieLesen(name){
    var arg =name+"=";
    var alen=arg.length;
    var clen=document.cookie.length;
    var i=0;
    while (i<clen){
      var j =i +alen;
      if (document.cookie.substring(i,j)==arg) return getCookieVal(j);
      i=document.cookie.indexOf(" ",i)+1;
      if (i==0) break;
    }
    return null;
  }

  function getCookieVal (offset){
    var endstr=document.cookie.indexOf(";",offset);
    if (endstr==-1) endstr=document.cookie.length;
    return unescape(document.cookie.substring(offset,endstr));
  }

  //-->
   </script>

</head>

<body bgcolor="white" text="black">
<div align="left">
<form name="hospital">
<select size="1" name="auswahl" onChange="changeH(this.form.auswahl.
options[this.form.auswahl.options.selectedIndex].value)">
  <option value="insel">Inselspital Bern</option>
  <option value="usz">Universit&auml;stsspital Z&uuml;rich</option>
  <option value="leuven">Universit&auml;tsspital Leuven</option>
  <option value="mayo">Mayo Medical Center</option>
</select>
</form>
```

```
<form name="group">
<select size="1" name="auswahl" onChange="changeG(this.form.auswahl.
options[this.form.auswahl.options.selectedIndex].value)">
  <option value="X">alle Berufsgruppen</option>
  <option value="A">An&auml;sthesie</option>
  <option value="AP">An&auml;sthesie Pflege</option>
  <option value="C">Chirurgie</option>
  <option value="LP">Lagerungsspezialist</option>
  <option value="P">OP_Pflegedienst</option>
</select>
</form>
</div>

  <script type="text/javascript" language="JavaScript">
  <!--
  var hosp= CookieLesen('hospital');
  var group= CookieLesen('group');
  for(i=0;i<document.group.auswahl.length;++i){
    if(document.group.auswahl.options[i].value == group)
      document.group.auswahl.options[i].selected=true;
  }
  for(i=0;i<document.hospital.auswahl.length;++i){
    if(document.hospital.auswahl.options[i].value == hosp)
      document.hospital.auswahl.options[i].selected=true;
  }
  // -->
  </script>

</body>
</html>
```