# Aufbereitung der Daten und Überführung in ein XML-Format

## 1) Teilautomatische Aufbereitung und Grundstruktur

### Vorverarbeitungsschritte

Die Basis für das Dortmunder Chat-Korpus bilden die Mitschnitte diverser Chats, so wie sie von den jeweiligen Chat-Anwendungen bezogen werden konnten. Die Ausgangsdaten sind im Falle serverseitig erzeugter Logfiles häufig im TXT-Format, im Falle clientseitig erzeugter Logfiles entweder HTML (bei Speicherung des Bildschirmsprotokolls über einen Browser) oder (bei der Copy & Paste-Sicherung des Bildschirmprotokolls) DOC bzw. RTF. Diese „Rohdaten“ wurden zunächst in ein HTML-Format überführt, sofern das Ausgangsformat nicht bereits HTML war. Manuell wurden anschließend irrelevante HTML-Annotationen entfernt. Durch eine Suchen & Ersetzen-Routine wurden des weiteren die einzelnen Chat-Beiträge automatisch als *messages* ausgezeichnet und die entsprechend modifizierten Dokumente als XML-Dateien gespeichert.

### Basale Modellierungseinheit: Der Chat-Beitrag

In Anbetracht der signifikanten Unterschiede zwischen der Organisation mündlicher Gespräche und der Handlungskoordination im Chat haben wir uns dafür entschieden, in den Korpusdokumenten nicht *Gesprächsbeiträge* ( *Turns* ), sondern *Chat-Beiträge* ( *messages* ) als Grundeinheiten unserer Modellierung anzunehmen. Unter einem *Chat-Beitrag* verstehen wir solche Teilnehmeräußerungen, die im Display aufgrund jeweils eines vorangehenden und eines nachfolgenden Absatzreturns als Einheiten isolierbar sind, die vom betreffenden Produzenten durch Ausführung eines Sendeakts als Einheit an den Chat-Server übermittelt und von diesem in das Display der Adressatenrechner

übermittelt wurden. Der *Chat-Beitrag* stellt somit eine lediglich formale Einheit dar; über seine Funktion oder den ihm von Seiten des Produzenten beigemessenen Handlungswert ist damit noch nichts ausgesagt.

### Teilautomatische Annotation

Mit dem selbst entwickelten Java-Werkzeug *Logfile2XML*[1] wurden die Dokumente nach Einfügung der *message* -Tags automatisch vorannotiert: Bestimmte Attribute und Attributwerte zum Element *message* wurden eingefügt sowie Emoticons und Asterisk-Ausdrücke unterhalb der *message* -Ebene ausgezeichnet. Das somit erzeugte rudimentäre XML-Format wurde in einem weiteren Schritt von Hand weiterbearbeitet, um automatisch nicht zweifelsfrei identifizierbare Elemente wie Adressierungen ( *address* ) oder im Text einer *message* erwähnte (nicht selten abgekürzte) Nicknames ( *nickname* ) auszuzeichnen und kleinere Unzulänglichkeiten der maschinellen Vorannotation nachzukorrigieren.

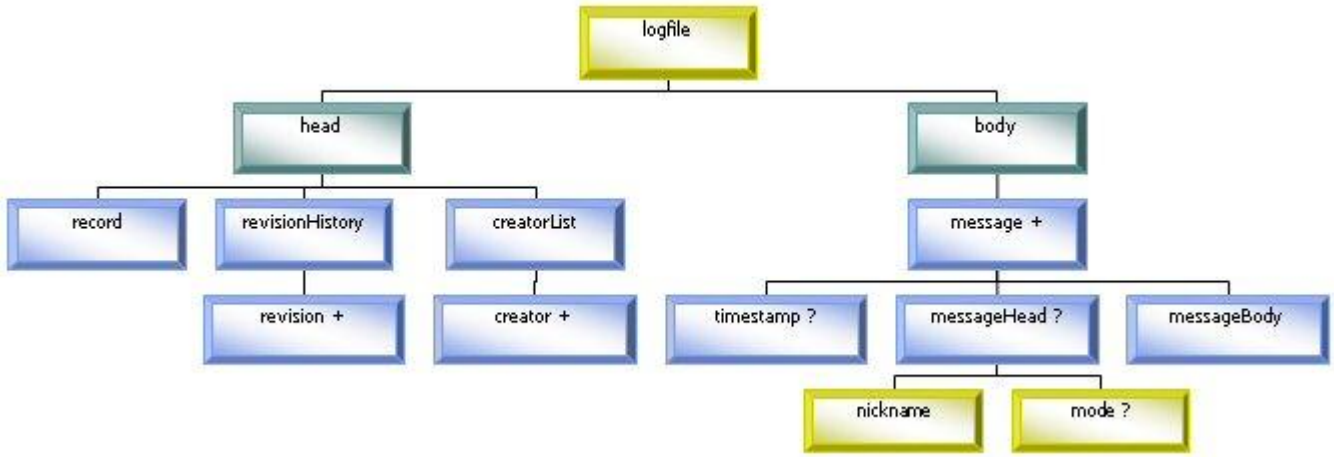
### Teilautomatische Generierung statistischer Daten

Die fertig annotierte Datei wurde zuletzt in das ebenfalls selbst entwickelte Java-Werkzeug *ExtendedHead* eingelesen. *ExtendedHead* generiert automatisch statistische Metadaten zum Inhalt eines Dokuments und vermerkt diese – ebenfalls in Form von XML-Annotationen – direkt im Dokument. Weitere Metadaten, die sich nicht maschinell auslesen lassen, für die Dokumentation der Korpusdaten jedoch wünschenswert waren, wurden anschließend von Hand ergänzt (z. B. das *estimatedGender* oder die *creatorList* ; s.u.).

[1] Alle für die Korpusaufbereitung verwendeten Programme wurden von *Bianca Selzam* entwickelt,

# 2) Das XML-Format im Überblick

Den XML-annotierten Korpusdokumenten liegt folgende Struktur zugrunde:




### Wurzelelement <logfile>

Das Wurzelelement *logfile* hat als Kinder *head* und *body* . Der *head* enthält Metadaten zum Mitschnitt, eine "Revision History" des Korpusdokuments sowie statistische Daten zum im Logfile dokumentierten Kommunikationsaufkommen. Im *body* hingegen steht mit den Chat-Daten der eigentliche Content des Mitschnitts.

### Element <head>

Die direkten Kindelemente von *head* sind *record* , *revisionHistory* und *creatorList* . Ihre Funktionen und Attribute werden im Folgenden erläutert.

### Element <head> :: Kindelement <record>

*record* enthält Informationen zum Chat-Angebot und dessen Aufzeichnung sowie statistische Daten.

- Attribut: *plattformName* - Wertebereich: Name des Chat-Angebots - obligatorisch
- Attribut: *plattformURL* - Wertebereich: URL des Chat-Angebots, z.B. http://www.unicum.de/chat - obligatorisch
- Attribut: *recDate* - Wertebereich: Aufzeichnungsdatum in der Form YYYY-MM-DD, z.B.: "2003-09-21" für "21. September 2003" (sofern bekannt, ansonsten „unknown“) - obligatorisch
- Attribut: *recStart* - Wertebereich: Starzeitpunkt der Aufzeichnung in der Form HH-MM, z.B. "19-25" für "19 Uhr 25 Minuten" (sofern bekannt, ansonsten „unknown“) - obligatorisch

- Attribut: *recEnd* - Wertebereich: Starzeitpunkt der Aufzeichnung in der Form HH-MM, z.B. "21-17" für "21 Uhr 17 Minuten" (sofern bekannt, ansonsten „unknown“) - obligatorisch
- Attribut: *recBy* - Wertebereich: Name des/der Aufzeichnenden (sofern bekannt, ansonsten "unknown") - obligatorisch
- Attribut: *TNOM* ("total number of messages") – Wertebereich: Anzahl der messages im Dokument-Body - obligatorisch
- Attribut: *TNOT* ("total number of tokens") – Wertebereich: Anzahl der laufenden Wortformen im Logfile - obligatorisch
- Attribut: *view* - Wertebereich: Nickname des Chatters, dessen Sicht auf das Kommunikationsgeschehen im Logfile dokumentiert ist (nur, sofern relevant) - fakultativ

### Element <head> :: Kindelement <revisionHistory>

Die *revisionHistory* dokumentiert die verschiedenen Aufbereitungs- und Bearbeitungsschritte eines Dokuments. Sie wird bei jedem Bearbeitungsvorgang bzw. bei jeder Änderung des Dokuments aktualisiert.

Das Element *revisionHistory* enthält beliebig viele Kindelemente des Typs revision.

### Element <head> :: Kindelement <revisionHistory> :: Kindelement <revision>

*revision* dokumentiert einen Bearbeitungsvorgang am Dokument. Die Elemente *revision* sind laufend durchnummeriert. Inhalt jedes *revision* -Elements ist eine Kurzbeschreibung der jeweils vorgenommenen Änderungen bzw. Überarbeitungsschritte.

- Attribut: *no* - Wertebereich: Laufende Nummer des Vorkommens des Elements *revision* - obligatorisch
- Attribut: *by* - Wertebereich: Name des Bearbeiters/der Bearbeiterin - obligatorisch

ANNOTATIONSBEISPIEL <revision>:

```
<revision no="6" by="Bianca Selzam">
      Metadaten hinzugefügt, Durchnumerierung der
      messages vorgenommen
</revision>
```

### Element <head> :: Kindelement <creatorList>

Die *creatorList* enthält eine Liste sämtlicher im Mitschnitt aktiver Chatter, die in beliebig vielen Kindelementen vom Typ *creator* kodiert sind.

### Element <head> :: Kindelement <creatorList> :: Kindelement <creator>

Ein Element *creator* repräsentiert einen Chatter in einem Logfile, der mindestens einen Beitrag aktiv verfasst hat.

- Attribut: *name* - Wertebereich: Nickname des Chatters bzw. "system" für das System (im Falle, dass das Logfile systemgenerierte Beiträge enthält) – obligatorisch
- Attribut: *estimatedGender* - Wertebereich: geschätztes Geschlecht: "male" / "female" / "unknown" / "system" - obligatorisch
- Attribut: *NOM* ("number of messages") – Wertebereich: Anzahl der vom betreffenden Chatter produzierten Messages (also derjenigen messages im Body, für welche er als creator fungiert) - obligatorisch
- Attribut: *NOT* ("number of tokens") – Wertebereich: Anzahl der laufenden Wortformen sämtlicher messages des betreffenden Chatters - obligatorisch
- Attribut: *role* - Wertebereich: Angabe einer Kommunikantenrolle (falls relevant), z.B. "moderator", "celebrity" - fakultativ

ANNOTATIONSBEISPIEL <creatorList>:

```
<creatorList>
      <creator
            name="system" estimatedGender="system"
            NOM="8" NOT="48"/>
      <creator
            name="Martin" estimatedGender="male"
            NOM="3" NOT="31"/>
      <creator
            name="Nina" estimatedGender="female"
            NOM="7" NOT="64"/>
      <creator
            name="Ludwig" estimatedGender="male"
            NOM="15" NOT="122" role="moderator"/>
</creatorList>
```

### Element <body>

Der *body* der XML-Datei enthält den eigentlichen Mitschnitt. Er enthält beliebig viele Elemente vom Typ *message* .

### Element <body> :: Kindelement <message>

Die Kategorie *message* beschreibt solche Einheiten, die von einzelnen Chattern durch Ausführung einer Verschickungshandlung (z.B. durch Betätigen der Eingabetaste oder Mausklick auf einen Sendebutton) an den Chat-Server aufgegeben wurden und die in den Logfiles jeweils einzeln als Produkte eines

bestimmten Urhebers ausgewiesen werden (in aller Regel durch automatische Voranstellung des Teilnehmer-Nicknames sowie durch vorangehenden und nachfolgenden Absatzreturn).

*messages* sind in den Mitschnitten durch vorangehende und nachfolgende Absatzreturns von einander abgegrenzt. Sie repräsentieren *Chat-Beiträge* .

- Attribut: *id* - Wertebereich: Laufende Nummer der *message* - obligatorisch
- Attribut: *type* - Wertebereich: Zuweisung eines der Subtypen *utterance * ("Äußerungsbeiträge"), *action* ("Beiträge mit Zuschreibungscharakter") oder *system * ("Systemmeldungen"). In den IRC-Chats tritt zusätzlich der Typ *bot* auf ("automatisch generierte Beiträge eines Chat-Robots") - obligatorisch
- Attribut: *creator* - Wertebereich: Name des Produzenten der *message* . Bei Systemmeldungen ist "system" angegeben - obligatorisch
- Attribut: *color* - Wertebereich: Farbwertangabe zur Anzeigefarbe der *message* - fakultativ

### Element <body> :: Kindelement <message> :: Kindelement <timestamp>

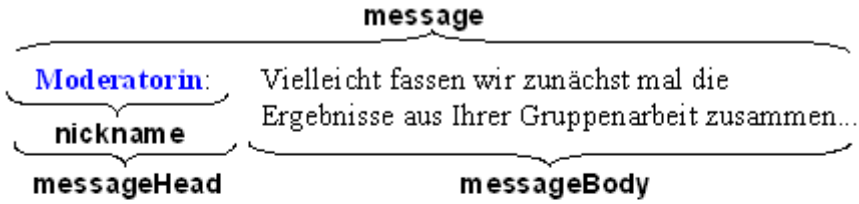
Dieses (fakultative) Element gibt den Zeitpunkt der Entgegennahme eines Beitrags durch den Chat-Server an. Der timestamp kann entweder message-initial oder message-final stehen:

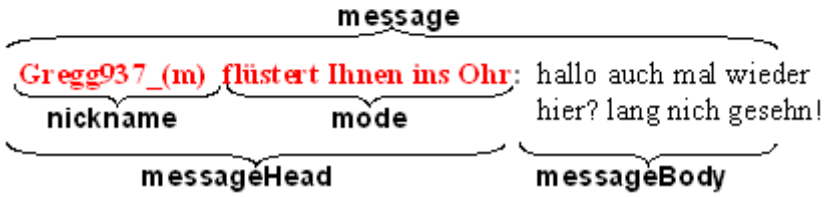


### Element <body> :: Kindelement <message> :: Kindelement <messageHead>

Der *messageHead* umfasst diejenigen Teile einer *message* , die (a) vom System automatisch generiert wurden und (b) die Funktion haben, den Produzenten des Beitrags anzuzeigen sowie ggf. den Äußerungsmodus zu benennen, der vom Produzenten für den Beitrag gewählt wurde (z.B. "Flüster"-Modus).

Kindelemente sind *nickname* und *mode* .




Element <body> :: Kindelement <message> :: Kindelement <messageBody>

Eine *message* beinhaltet immer einen *messageBody* . Dieser umfasst denjenigen Teil der *message* , der die vom betreffenden Teilnehmer eingegebene Zeichenfolge (und somit den "Beitrag" im engeren Sinne) wiedergibt.


Adressierungen innerhalb des Elements *messageBody* werden durch das Kindelement *address* gekennzeichnet. Das zugehörige Attribut *addressee* zeigt, welcher andere Chat-Teilnehmer als Adressat gewählt wurde. Da Nicknames in Adressierungen häufig abgekürzt werden (z.B. „anton“ anstatt „anton23“) und bisweilen aus Flüchtigkeit auch Tippfehler enthalten („atnon“ statt „anton“), ist die Belegung des Attributs *addressee* obligatorisch. Während das Element *address* diejenige Zeichenfolge markiert, die im Beitrag als Adressierung fungiert, wird als Wert zu *addressee* die originäre Form des Nicknames angegeben.

ANNOTATIONSBEISPIEL <address> im <messageBody>:

```
<message
      id="20" type="utterance" creator="tourteam"
      color="#CC0000">
      <timestamp>
            17:02:06
      </timestamp>
      <messageHead>
            <nickname>tourteam</nickname>
      </messageHead>
      <messageBody>
            <address
                  addressee="anton23">atnon:
            </address>
            wir müssen und noch ein paar tage gedulden
      </messageBody>
</message>
```

Wenn andere Chatter nicht direkt adressiert, sondern im Rahmen eines Teilnehmerbeitrags erwähnt werden, so werden diese Erwähnungen im *messageBody* mit dem Kindelement *nickname* ausgezeichnet, und zwar zunächst unabhängig davon, ob der betreffende Teilnehmer tatsächlich mit seinem Nickname, mit seinem realweltlichen Namen oder einem anderen sprachlichen Ausdruck genannt wird (z.B. „Torsten“ anstelle von „Rocky19“ oder „Grenzwall“ anstelle von „Limes“). Falls der verwendete Ausdruck vom Nickname abweicht, ist zum Attribut *baseform* der im Chat verwendete Nickname des betreffenden Teilnehmers als Wert angegeben.

ANNOTATIONSBEISPIEL <nickname> im <messageBody>:

```
<message
      id="339" type="utterance" creator="quaki"
      color="#D62994">
      <messageHead>
            <nickname>quaki</nickname>
      </messageHead>
      <messageBody>
            <nickname baseform="limes">
            der grenzwall
            </nickname>
            is schon wieda da heheh
      </messageBody>
</message>
```

„Netspeak“-Elemente wie Emoticons und Handlungsbeschreibungen in Asterisken werden durch die Tags *emoticon* und *asteriskExpression* markiert. Diese können auch ineinander verschachtelt sein.

ANNOTATIONSBEISPIEL <emoticon> / <asteriskExpression> im <messageBody>:

```
<message
      id="599" type="utterance" creator="TomcatMJ"
      color="#003388">
      <messageHead>
            <nickname>TomcatMJ</nickname>
      </messageHead>
      <messageBody>
            <asteriskExpression>
            *hängematte in baum aufspann
            und mich reinleg*
            </asteriskExpression>
            <emoticon>:-)</emoticon>
      </messageBody>
</message>
```

Michael Beißwenger / Bianca Selzam (*2005*)