Dialog Coding —
Function and Grammar

Göteborg Coding Schemas

Ed. Jens Allwood

GPTL

Gothenburg Papers in Theoretical Linguistics

Papers in GPTL cover general and theoretical topics in linguistics. The series contain papers from single individuals and project groups. It appears irregularly in two subseries, blue for English Papers, and green for Swedish Papers.

JENS ALLWOOD
Editor, GPTL

PREFACE

This volume contains coding schemas which pertain to the analysis of spoken language interaction and dialog. The schemas developed have been inspired both by theory and empirical work with spoken language corpora. Theoretical inspiration has come from many sources, the most important being speech act theory, Wittgenstein's philosophy of language, Conversational Analysis, and Activity Based Communication Analysis. Empirical experience with the schemas has come from work on the Göteborg Corpus of spoken language over a period of twenty years. The following types of coding are covered in the volume:

1. Social Activity and Communicative Act-related Coding
   - Social activity
   - Communicative acts
   - Expressive and Evocative functions and Obligations
2. Communication Management-related Coding
   - Feedback
   - Addressee, turn, and sequence management
   - Own Communication Management
3. Grammatical Coding
   - Parts of speech (automatic, probabilistic)
   - Maximal grammatical units

PART I

SOCIAL ACTIVITY AND COMMUNICATIVE ACT-RELATED CODING

SOCIAL ACTIVITY CODING

1. Description of Social Activity Coding

Each transcription is linked to a database entry and a “header” containing information on

a) the purpose, function and procedures of the activity,

b) the roles of the people participating in the activity,

c) the artefacts, i.e., objects, furniture, instruments and media of the activity,

d) the social and physical environment, and

e) anonymous categorical data on the participants, such as age, gender, dialect and ethnicity.

In addition, the major subactivities of each activity are given.

Below is an example of what a “header” for a transcription of an “auction” looks like. This is part of the activity coding.

@ Recorded activity ID: A791901
@ Recorded activity date: 960309
@ Recorded activity title: Auktion II
@ Short name: Auction II
@ Tape(s): A7919
@ Participant: A = Auctioneer
@ Participant: G = Gustav
@ Participant: U = Unknown girl
@ Transcription name: A7919011
@ Transcription System: MSO6
@ Duration: 01:37:47
@ Transcriber(s): NN NN
@ Transcription date(s): 980116, 981005
@ Transcribed segments: All
@ Checker(s): XX XX
@ Checking date(s): 981126
@ Time coding: Yes
@ Section: 01. Start
@ Section: 02. L 78:87 Buffet
@ Section: 03. S 89:111 Persian Hamadan rug
@ Section: 04. S 89:84 Mirror
@ Section: 05. L 78:90 Painted cupboard

In order to give a better idea of what an acitivity coding involves we will now give five (5) examples of activity type codings—”auction”, “medical consultation”, “dinner”, “formal meeting”, and “information service”.

2. Examples of Social Activity Coding

2.1 Activity coding for an AUCTION

<table>
<tr><td rowspan="2"><b>PURPOSE</b>
Selling goods by exhibiting them singly and bidding</td><td colspan="3"><b>Activity structure</b></td><td colspan="4"><b>Subgoals</b></td><td colspan="2"><b>Procedures</b></td></tr>
<tr><td colspan="3">Every sold item delimits a subactivity, which normally consists of:
1. Presentation of the item
2. Bidding
3. Determine buyer and price of the item</td><td colspan="4">4. Determine buyer and price of each item</td><td colspan="2">Every item is first presented with a description or with its number. Then the auctioneer tries to get someone to bid at the start bid. He then encourages the audience to bid higher prices. When nobody can bid higher, the bidder who bid the highest price gets to buy the item at that price.</td></tr>
<tr><td rowspan="3"><b>ROLES</b></td><td></td><td colspan="4"><b>Competence</b></td><td colspan="3"><b>Rights</b></td><td><b>Obligations</b></td></tr>
<tr><td>Auctioneer</td><td colspan="4">Knowledge of the routines during an auction</td><td colspan="3">Determine buyer and price from the bidding</td><td>Reliability—listen to the right bidder and the right price
Choose the buyer who bids at the highest price</td></tr>
<tr><td>Bidders</td><td colspan="4"></td><td colspan="3">Bid and buy
Get reliability from auctioneer—he should listen to the right bidder and the right price</td><td>Let auctioneer determine buyer and price</td></tr>
<tr><td rowspan="2"><b>ARTIFACTS</b></td><td colspan="2"><b>Artifacts</b></td><td colspan="4"><b>Instruments</b></td><td colspan="3"><b>Media</b></td></tr>
<tr><td colspan="2">Goods to be sold</td><td colspan="4">Auctioneer’s hammer and something to hit it against</td><td colspan="3">Direct Speech
(audio or video recorder used for recording purposes)</td></tr>
<tr><td rowspan="2"><b>ENVIRON-MENT</b></td><td colspan="4"><b>Social–Cultural</b></td><td colspan="5"><b>Physical</b></td></tr>
<tr><td colspan="4">Auctioneer probably doesn’t know most of the bidders/audience</td><td colspan="5">Outside or in a big room
Auctioneer placed in front of and facing the audience/ possible bidders
Tape recorder used for the recording</td></tr>
</table>

2.2 Activity coding for MEDICAL CONSULTATION

PURPOSE	Activity structure	Subgoals	Procedures
Consultation by physician to help/cure the patient	1. Greeting 2. Diagnosis or check-up 3. Writing of prescription if needed	4. If first visit: Make a diagnosis for the patient 5. Determine treatment	The physician and the patient greet each other. If it is the first visit, the physician tries to diagnose by asking questions and/or making a physical examination. Then he determines treatment. If the patient has been there before, the physician discusses the patient’s condition with the patient. Then the physician decides whether to continue with the current treatment or not.

ROLES		Competence	Rights	Obligation
	Physician	Medical degree	Ask questions about patient’s condition Make physical examination Diagnose Write prescriptions	Reliability—help the patient in the best possible way Professional secrecy
	Nurse	Nurse education	Ask questions about patient’s condition Make physical examination	Reliability—help the patient in the best possible way Professional secrecy
	Patient	-	Get advice, and prescription if needed Sincerity from physician and nurse Get help in the best possible way	Reliability about condition and symptoms
	Relatives of patient	-	-	Sincerity about condition and symptoms

ARTIFACTS	Instruments	Media
	Medical instruments	Direct speech (audio or video recorder used for recording purposes)

ENVIRONMENT	Social–Cultural	Physical
	Physician/Nurse and patient probably don’t know each other	Hospital or other surgery

2.3 Activity coding for informal DINNER

<table>
<tr><td rowspan="2"><b>PURPOSE</b>
Have dinner and informal conversation</td><td colspan="2"><b>Activity structure</b></td></tr>
<tr><td colspan="2">1. Serving dinner
2. Eating and drinking
3. Conversation during the meal</td></tr>
<tr><td><b>ROLES</b></td><td colspan="2">Participants in the meal</td></tr>
<tr><td rowspan="2"><b>ARTIFACTS</b></td><td><b>Instruments</b></td><td><b>Media</b></td></tr>
<tr><td>Food
Drink
Cutlery
China
Table
Chairs</td><td>Direct speech
(video or tape recorder used for recording purposes)</td></tr>
<tr><td rowspan="2"><b>ENVIRONMENT</b></td><td><b>Social–Cultural</b></td><td><b>Physical</b></td></tr>
<tr><td>Most or all of the participants usually know each other.</td><td>Place: Somebody’s home
People are sitting around a table</td></tr>
</table>

2.4 Activity coding for FORMAL MEETING

	Activity structure	Subgoals	Procedures
PURPOSE	Every issue being discussed defines a subactivity, normally consisting of: 1. Definition of the problem or issue or following up on an earlier issue 2. Discussion or try to solve the problem 3. Delegating	4. Try to solve problems that arise 5. Delegate assignments	If there is an agenda, it is followed and issues are being discussed in that order. Otherwise, the chairman usually gives the word to the different participants. Every decision should be written in the protocol by the secretary.

		Competence	Rights	Obligations
ROLES	Chairman	Familiarity with routines during a formal meeting	Lead the meeting	Lead the meeting Let everyone talk who wants to Listen to problems/opinions
	Project/ department employees		Express problems and opinions	Wait for turn
	Secretary	Knowledge how to write a protocol	Ask for clarification or specification	Write a protocol

	Instruments	Media
ARTIFACTS	Agenda (poss.) Protocol	Direct speech (video or tape recorder used for recording purposes)

	Social–Cultural	Physical
ENVIRONMENT	If this is a meeting in a workplace, most of the meeting participants know each other and are probably workmates. In other meetings there is often a lack of familiarity between participants.	Meeting room at work Participants are seated around a table

2.5 Activity coding for INFORMATION SERVICE (PHONE)

<table>
<tr><td rowspan="2"><b>PURPOSE</b></td><td colspan="2"><b>Activity structure</b></td><td colspan="3"><b>Subgoals</b></td><td colspan="2"><b>Procedures</b></td></tr>
<tr><td colspan="2">1. Caller asks for information
2. Information giver tries to give the information wanted.</td><td colspan="3">3. Make sure that the information giver has understood correctly and that he/she gives correct information</td><td colspan="2">The caller phones and when the call is answered, he/she makes an inquiry, which the information giver should try to answer.</td></tr>
<tr><td rowspan="3"><b>ROLES</b></td><td></td><td colspan="2"><b>Compe-tence</b></td><td colspan="3"><b>Rights</b></td><td><b>Obligations</b></td></tr>
<tr><td>Caller</td><td colspan="2"></td><td colspan="3">Ask relevant questions
Be treated politely
Get relevant, correct information</td><td>Politeness</td></tr>
<tr><td>Information giver</td><td colspan="2">Knowledge within the area the call is about</td><td colspan="3">Be treated politely</td><td>Give relevant and correct information
Politeness</td></tr>
<tr><td rowspan="2"><b>ARTIFACTS</b></td><td colspan="4"><b>Instruments</b></td><td colspan="3"><b>Media</b></td></tr>
<tr><td colspan="4">Computers are sometimes used to find information</td><td colspan="3">Phone
(tape recorder used for recording purposes)</td></tr>
<tr><td rowspan="2"><b>ENVIRONMENT</b></td><td colspan="5"><b>Social–Cultural</b></td><td colspan="2"><b>Physical</b></td></tr>
<tr><td colspan="5">Formal—
Caller and
Information giver don’t know each other</td><td colspan="2">The conversation takes place over the phone.</td></tr>
</table>

COMMUNICATIVE ACTS

Coding Manual

Jens Allwood, Elisabeth Ahlsén, Maria Björnberg, and Joakim Nivre
Version 1, 2000-01-04

1. Communicative Acts

Below is a summary of the Communicative Acts found in the travel agency dialog “Flyg till Paris”.

Compressed Communicative Acts	**Speech Act Labels**
Acceptance	Acceptance Acceptance of task
Acknowledgement	
Affirmation	
Agreement	
Answer	Answer Initiated answer Continued Answer
Check of comprehension	
Clarification	
Confirmation	
Conclusion	Initiated conclusion Continued conclusion
Elaboration of objection	
Elicitation of agreement	
Ending interaction	
Excuse	
Explanation	Explanation Explanation of conditions for discount
Hesitation	
Interruption	
Joke	
Keep turn	

Objection	Objection Hesitating objection
Offer	
Providing alternative flight	
Qualification	
Question	Question Initiated question Continued question
Reformulation	
Reminder	
Repetition	
Request	Request for alternative cheap traveling Request for clarification Request for contact Request for info about discount Request for info about discount traveling Request for information Request for low price ticket Request for specification of eligibility of discount Request for specification of traveling time Request for specification of type of ticket Request to wait
Self confirmation	
Self introduction	
Specification	Specification of price Specification of price range Specification of traveling time Specification Initiated specification Continued specification

Statement	Statement Initiated statement Continued statement Statement of main information need Statement of main task
Summons	
Unclear	

2. Example coding

This is an example of speech act coding, using the speech act labels in the preceding chapter. A slash (/) indicates that the utterance has multi-functionality and has been coded with more than one speech act label. A plus (+) separates speech act codings for different parts of the utterances.

Dialog	**Speech acts**
$P1: hup	Summons/Request for contact
$J1: [1 {j}a:]1	Acceptance (P1)
$P2.1: [1 ö:m]1 //	Hesitation/Keep turn +
$P2.2: flyg ti{ll} <1 paris >1	Request for information/Statement of main task/Statement of main information need
@ <1 name >1	
$J2.1: mm <2 >2 <3 /	Acceptance of task(P2.2)
$J2.2: ska [2 du ha:]2 en returbiljett >3	J2.2a:Question/J2.2b:Request for specification of type of ticket
@ <2 event: P opens her bag >2	
@ <3 event: people are talking in the background >3	
$P3: [2 ö:{h}]2	Hesitation
$P4: va{d} sa du	P4a:Request for clarification(J2)/P4b:Question
$J3: ska du ha en tur å0 retur	Answer(P4b)/Clarification(J2)/Repetition (J2)/Question/Request for specification of type of ticket
$P5.1: ja <4 / >4	Answer(J3)/Specification(J3)
$P5.2: ö{h}	Hesitation
@ <4 inhalation sound (burping): J >4	
$J4: // vi{l}ken månad ska du åka	Question/Request for specification of traveling time
$P6.1: / <5 <6 >5 >6 ja:	Hesitation +

$P6.2: typ den:	Initiated (answer(J4)/Statement/ Specification(J4))
$P6.3: ä:{h}	Hesitation
$P6.4: tredje fjärde <7 <8 april >7 / [3 nån]3 gång där > 8 <9 / >9	Continued(answer(J4)/Statement/ Specification(J4))
$P6.5: så billi{g}t [4 som möjli{g}t]4	Statement/Specification of price range/ Request for low price ticket
@ <5 sigh >5 @ <6 event: P is looking through some papers >6 @ <7 name >7 @ <8 puffing >8 @ <9 inhalation sound: J >9	
$J5: [3 mm]3	Acceptance(P6.2)
$J6.1: <10 [4 ja just]4 de{t} jo /	Acceptance(P6.4)/Reminder
$J6.2: de{t} ha{r} ja{g} aldri{g} hört förr /	Statement
$J6.3: de{t} billi{g}aste >10 vi har <11 e:0 >11 <12 air france >12 ettusenåttahundratie / [5 plus]5 flygplatsskatter	Statement/Specification of price
$J6.4: så du hamnar pÅ:	Initiated conclusion/Initiated statement +
$J6.5: <13 >13 {j}a du kan få exakt	Offer +
$J6.6: <14 vänta ska du se här vi gö{r} såhär	Request to wait
$J6.7: / ö:{h} // >14	Hesitation
@ <10 giggling: P >10 @ <11 inhalation sound: P >11 @ <12 name >12 @ <13 inhalation sound >13 @ <14 event: J is typing on a computer keyboard >14	
$P7: [5 {j}a:]5	Acceptance(J6.3)
$J7.1 : de{t} är en skatt i: $J7.2 : e{h} / $J7.3 : <15 >15 bÅde <16 danmark >16 och i <17 frankrike >17 $J7.4 : så du ska få <18 exakt / $J7.5 : >18 <19 se{da}n mÅste du ha e{tt} sån där inte{r}nationellt studentkort också $J7.6: ha{r} du de{t} >19	Statement + Hesitation + Continued statement (J7.1)+ Offer + Statement + J7.6a:Question/J7.6b:Request for specification of eligibility of discount

@ <15 click >15 @ <16 name >16 @ <17 name >17 @ <18 event: J is typing on a computer keyboard >18	
@ <19 quick >19	
$P8.1: <20 mm $P8.2: nä: >20	Hesitation + Answer(J7.6a)
@ <20 event: P is going through some papers >20	
$J8: <21 / du vet va{d} de{t} e0 fö{r} nåt ja{g} syfta{r} på >21	Question/Check of comprehension
@ <21 event: J is typing on a computer keyboard >21	
$P9: {j}a:	Answer(J8)
$J9.1: <22 då ska vi > $J9.2: 22 de{t} kosta{r} nitti{o} kroner om du inte har de{t}	Initiated conclusion + Statement
@ <22 quiet >22	
$P10: <23 mm >23 / <24 >24	Acceptance(J9.2)
@ <23 quiet >23	
@ <24 inhalation sound >24	
$J10.1: då ska vi se / $J10.2: <25 >25 ö:{h} med skatter tvåtusensextio / [6 <26 köpenhamn >26 <27 paris >27]6	Initiated conclusion + Statement/Continued conclusion
@ <25 inhalation sound >25	
@ <26 name >26	
@ <27 name >27	
$P11.1: [6 <28 oke:j >28]6 $P11.2: / dÅ e0 de{t} från <29 köpenhamn >29 <30 å0 [7 så]7 >30	Acceptance(J10.2) + Question
@ <28 loan english: okey >28 @ <29 name >29 @ <30 mumbling >30	
$J11: [7 {j}a]7 just de{t}	Affirmation(p11.2)
$P12: mm	Confirmation(J11)
$J12.1: vi ha{r} ju <31 äf+ >31 $J12.2: vi ha{r} ju <32 sas >32 också $J12.3: då få{r} du en från <33 mal:mö: >33 / $J12.4: {j}a vi ska se: $J12.5: <34 tjuge <35 >35 noll noll (...) femti{o} >34 /	Statement + Reformulation(J12.1)/ Continued statement (J12.1) + Statement + Request to wait + Continued statement(J12.3) +
$J12.6: ska vi se va{d} de{t} blir //	Request to wait +

$J12.7: just här måste du ta dej till / <36 köpenhamn >36 / [8 pÅ den]8	Statement/Providing alternative flight
@ <31 cutoff: Air France >31, <31 name >31	
@ <32 acronym >32	
@ <33 name >33	
@ <34 mumbling >34	
@ <35 event: the phone starts ringing >35	
@ <36 name >36	
$P13: [8 {j}a:]8	Acceptance(J12.7)
$J13.1: <37 då ska vi se >37 $J13.2: <38 malmö: >38 / tvåtusenfyrahundrafyrti{o}fem	Request to wait + Statement/ Specification of price
ß Answering the phone / ending the conversation	
@ <37 quick >37	
@ <38 name >38, <38 slow >38	
# 00:01:42	
$J14: /// <39 >39 [9 ja{g} ska bara+]9	Request to wait
@ <39 inhalation sound >39	
$P14: [9 å0 då behöve{r}]9 man inte köpa nåt sånt e{h} [10 inte{r}nationellt]10	Question/Request for info about discount
$J15: [10 jo]10	Answer(P14)/Objection(P14)
$P15: (behöve{r} man)	Question
$J16.1: ja{g} ska bara be dom å1 dröja $J16.2: vänta lite <40 >40 <41 sta >41	Statement + Request to wait +
$J16.3: <42 johannes >42 $J16.4: kan ni dröja ett kort ögonblick bara <43 / >43	Self introduction + Request to wait +
$J16.4: [11fö{r}låt]11	Excuse
@ <40 event: J answers the phone >40	
@ <41 abbreviation >41	
@ <42 name >42	
@ <43 click: P >43	
$P16.1: [11 ha]11 $P16.2: så: e0 [12 de:{t}]12	Acceptance(J15) + Question
$J17: [12 jo du]12 måste ha ett sånt kort till e{h} / ett sådant	Continued Answer(P14)/Interruption/ Statement/Elaboration of objection(P14) / Explanation of conditions for discount

$P17: mm	Acknowledgement(J17)
$J18.1: ti{ll} bägge dom hä{r} $J18.2: va	Continued Statement (J17) + Elicitation of agreement with J18.1
$P18.1: mm / $P18.2: [13 jo de{t}]13	Agreement(J18.1)/Acceptance(J18.1) + Initiated statement
$J19: [13 fö{r} de{t}]13 e0 studentbiljett du kan boka nä{r} du vill du kan ändra datum på	Interruption/Statement/ Explanation(J17)
$P19.1: mm å0 {h}m {j}a precis $P19.2: då $P19.3: e{h} $P19.4: kan man åka / [14 nä{r}]14 som helst	Acceptance(J19) + Initiated conclusion + Hesitation + Continued conclusion (P19.2)/Question/Request for info about discount traveling
$J20: [14 fö{r}låt]14	Interruption/Excuse
$J21.1: {j}aa $J21.2: bara du e0 hemma inom ett år <44 / mm >44	Answer(P19.4)/Confirmation(J19) + Statement
@ <44 giggle: P >44	
$P20.1: <45 [15 {j}a {j}a]15 $P20.2: de:{t} >45	Acceptance(J21) + Initiated statement
@ <45 giggling >45	
$J22: [15 mm <46 {j}a >46]15	Confirmation(P20.1)
@ <46 ingressive >46	
$P21.1: så de{t} finns inga andra biljetter som e0 $P21.2: de{t} e0 de{t} billi{g}aste	Initiated question + Question
$J23: nae de{t} e0 de{t} billi{g}aste ja{g} har <47 air france >47 de{t}	Answer(P21.1)/Statement/ Affirmation(P21.2)
@ <47 name >47	
$P22: <48 hap >48 / <49 okej >49	Acceptance(J23)
@ <48 puffing >48, <48 SO: jaha >48 @ <49 loan english: okey >49	
$J24: mm	Confirmation(P22)
$P23.1: <50 men e{h} >50 $P23.2: {j}a ja{g} ska <51 >51 / <52 {j}a >52	Hesitating objection + Statement
@ <50 giggling >50 @ <51 inhalation sound >51 @ <52 quiet >52	
$J25: [16 (hemma direkt)]16	Unclear
$P24: [16 men de{t} går]16 väl / flyg hela tiden anta{r} ja{g} a{llt}så	Question/Request for info about discount traveling

$J26.1: e{h} $J26.2: [17 ja:a <53 air france >53]17 ha{r} ju fyra plan om da{ge}n å0 [18 <54 sas >54 ha{r}]18 nå{got} liknande	Hesitation + Answer (P24)/Statement
@ <53 name >53 @ <54 acronym >54, <54 name >54	
$P25: [17 i stort sett]17	Continued question
$P26: [18 {j}aa:]18	Acceptance(J26.2)
$P27: mm <55 okej >55	Acceptance(J26.2)
@ <55 loan english: okey >55	
$J27: men påsken e0 ju ganska svår	Statement/Qualification(J26)
$P28: <56 >56 {j}a [19 jo]19	Acceptance(J27)
@ <56 inhalation sound >56	
$J28: [19 mm]19	Confirmation(P28)
$J29: <57 {j}a >57	Confirmation(P28)
@ <57 ingressive >57	
$P29.1: <58 okej >58 / <59 >59 $P29.2: [20 {j}a men+]20	Acceptance(J26–J28) Self confirmation(P29.1)
@ <58 loan english: okey >58	
@ <59 inhalation sound >59	
$J30.1: [20 du få{r}]20 fundera lite $J30.2: du e0 välkommen igen	Statement + Offer/Statement
$P30: {j}a: <60 >60	Acceptance(J30)/Ending interaction
@ <60 giggle >60	
$J31: mm	Confirmation(J30,P30)/Ending interaction

EXPRESSIVE & EVOCATIVE FUNCTIONS

AND OBLIGATIONS

Coding Manual

Jens Allwood

Version 1.0   January, 2000

1. Contributions, expressive, and evocative functions

1.1 Contributions

Following Grice (1975), Allwood, Nivre, and Ahlsén (1990), and Allwood (1995), the basic units of dialog are gestural or vocal *contributions*1 from the participants. The term *contribution* is used instead of *utterance* in order to cover also gestural and written input to communication. Verbal contributions can consist of single morphemes or be several sentences long. The term *turn* is used to refer to the right to contribute, rather than to the contribution produced during that turn. One may make a contribution without having a turn and one may have the turn without using it for an active contribution, as demonstrated in the example below, in which B's first contribution involves giving positive feedback without having the turn (square brackets indicate overlap) and his second contribution involves being silent and doing nothing while having the turn.

```
A: look ice cream [would] you like an ice cream
B1:                      [yeah]
B2: (silence and no action)
```

1.2 Expressive and Evocative Function

In accordance with Allwood (1976, 1978, 1995), each contribution is viewed as having both an *expressive* and an *evocative* function. The *expressive* function lets the sender express beliefs and other cognitive attitudes and emotions. What is "expressed" is made up of a combination of reactions to the preceding contribution(s) and novel initiatives. The *evocative* function is the reaction the sender intends to call forth in the hearer. Thus, the evocative function of a statement normally is to evoke a belief in the hearer, the evocative function of a question is to evoke an

[1] The term *contribution* has been used in various ways. Clark and Schaeffer (1989), use the term in a more restricted sense to refer to what they call "grounded" contributions. They use the term *presentations* for single agent contributions that may or may not have been "grounded".

answer, and the evocative function of a request is to evoke a desired action. For a discussion of the relations between these functions and Bühler's symptom, symbol, and signal function (1934), as well as Austin's locutionary, illocutionary, and perlocutionary functions (1962), see Allwood (1976, 1977, 1978). The notion of evocative function is also similar to the notion of "intended perlocutionary function" of Sadek (1991).

Each contribution to a dialog is associated with the following default evocative functions, cf Allwood (1987,1995). A contribution is intended to make the receiver:

(i) continue (C),
(ii) perceive (P),
(iii) understand (U), and
(iv) react in accordance with main evocative function (R).

The receiver now has to evaluate whether he or she can/wants to continue, perceive, understand, and go along with the evocative intention of the preceeding utterance. The result of the evaluation will be an important part of the expressive function of the response to this utterance and can be given in explicit or implicit form (see below). Using these concepts, we now turn to an analysis of the cooperative use of the expressive and evocative aspects of contributions. We can provide a more detailed analysis of the cooperative goal of communication into four subgoals, related to the four evocative/expressive functions, one of which is the joint understanding we have already discussed:

(i) Continued interaction until both parties agree to halt
(ii) Joint perception and awareness
(iii) Joint understanding
(iv) Cooperative achievement of evocative intentions

2. Obligations

If the four subgoals mentioned above are to be cooperatively pursued, whether it be in the service of some activity or not, they impose certain obligations on both sender and receiver. With regard to both expressive and evocative functions, the sender should take the receiver's perceptual, cognitive and behavioral ability into cognitive and ethical consideration and should not mislead, hurt or unnecessarily restrict the freedom of the receiver. The receiver should reciprocate with an evaluation of whether he/she can hear, understand and carry out the sender's evocative intentions and signal this to the interlocutor. Without reasons to the contrary, the sender and receiver should also trust the other to behave in this manner.
The sender's and receiver's obligations can be summarized as follows (see also Allwood 1994):

Sender:

1. *Sincerity*:
The sender should, unless she/he indicates otherwise, have the attitude normally associated with a particular type of communicative act, e.g. statement–belief, request–desire (cf. Allwood 1976, 1995).

2. *Motivation*:
Normally, communicative action, like other action should be motivated.
3. *Consideration*:
If communicative action is to be cooperative and ethical it must take the other person into cognitive and ethical consideration.

Receiver:
1. *Evaluation*:
The receiver should evaluate the preceding utterance with regard to whether he/she can continue the interaction, perceive and understand and accept its main evocative intention.
2. *Report*:
After having evaluated, the receiver should report the result verbally or nonverbally.
3. *Action*:
In some activities and roles, a positive evaluation of the ability to carry out the main evocative intention also obligates the listener to carry out the action associated with this intention.

Since perception and understanding mostly function as a means for the sharing of the expressive and evocative functions of each contribution, a cooperative response usually consists in one of the following responses, used separately or in combination:

(i) overtly signaling the result of the listener’s evaluation through the use of an explicit positive or negative feedback expression, such as a head nod, a head shake or a verbal expression like m, what, yes, no or *OK*, after a statement or request
(ii) direct verbal action, as when a question is answered
(iii) direct nonverbal action, as when a window is closed after a request to do so
(iv) implicitly accepting an evocative intention by contributing a response that implies acceptance, as when you accept a stated belief by exploring one of its consequences

Since the main thrust of a dialog revolves around evocative intentions which are aimed at achieving more than mere perception and understanding, a cooperative response that signals only perception and understanding usually occurs only in the following circumstances: (i) when a message can be perceived and understood but no commitment is made to its evocative function or (ii) a message cannot be perceived or understood. In the first case often low key feedback expressions like *m* or *well* are used and in the second we find instead negative feedback expressions such as *pardon* or *what*. These issues are explored further in Allwood, Nivre, and Ahlsén (1992).

3. Examples—Expressive and evocative functions and obligations

We now turn again to the travel dialog and the quarrel to illustrate what expressive and evocative functions and obligations might be involved in dialogs of these types. Every utterance, unless otherwise coded, either implicitly or explicitly expresses CPU (contact, perception, and understanding). CPU are only coded when they are part of the main evocative or expressive function of an utterance. When they are not, another expressed attitude such as acceptance or belief will imply CPU which therefore will not be indicated. Similarly, “acceptance” of information will only be indicated if it is part of what is mainly expressed. If a question is followed by an answer, the answer to the question will be taken to imply acceptance of the task of answering. If a request is followed by the required action, the action will be taken to imply acceptance of carrying out the task, and if a statement is followed by a comment which presupposes what is stated to be true, the comment will be taken to imply acceptance of the information expressed by the statement. In all these cases acceptance will not be coded. A comment is also needed about statements. Statements can be implied or explicit. Answers to questions often contain implicit statements. An answer to a yes/no question, for example, implies an affirmed or negated statement of what is queried in the question. If a statement is implicit, we will code its related expressive functional commitment as an expression of and a commitment to the propositional information in the statement. If it is explicit, we will code the statement as an expression of and a commitment to a belief containing that propositional information. In terms of commitments the two will be equivalent, but the former code has the advantage that an informational object can be shared between questions, answers, and requests. We ask for and request information rather than beliefs, even though what a conversational interaction will provide are beliefs containing such information.

The next step is to link the expressive and evocative functions with utterance and dialog act-related obligations, which can now be added as modifications of the role-related obligations we have discussed above. In coding obligations we will, for the speaker, normally indicate commitment to whatever attitude and motive that has been expressed by the speaker. Unless it is relevant, we will normally not indicate that the utterance also should be based on cognitive and ethical consideration of the listener. For the listener, the fundamental obligations are never more than evaluation and response (report), but if circumstances are such that a positive evaluation takes place and the role relation is such that the listener, *ceteris paribus*, is obliged to act in conformity with the speaker’s main evocative intentions, we will also, in brackets, indicate this action as part of the listener’s obligation. In such cases we will leave out the “respond” obligation which, in case the evaluation is negative, will become the main obligation.

In the tables below, “/” means simultaneous functions, “;” means functions occurring sequentially, + means linked obligations. Variables such as X and Y are used as shorthand for the actual information, and utterances are referred to by speaker and number.

3.1 The Travel agency dialog—Flight to Paris

Travel Dialog—Expressive and Evocative Functions and Obligations

Contribution	**Expressive and Evocative Function**	**Obligations Introduced**
C1: hup	expr: presence/desire for contact evoc: CP/start interaction	speaker: commitment to interest in contact listener: evaluate + respond
A1: [1a]1	expr: CPU Acc (C1) evoc: state request	speaker: commitment to contact listener: evaluate + respond
C2: [1öm]1 // flyg ti Paris:	expr: hesitation; desire for info [X] evoc: give info [X]	speaker: commitment to interest in info (X) listener: evaluate + (give info (X))
A2: mm/ ska [2 du ha]2 en returbiljett	expr: accept evoc [C2]; desire for info [Y] evoc. give info [Y]	speaker: commitment to need info (Y) listener: evaluate + (give info (Y))
C3: [2 ö] 2	expr: C hesitation evoc: C	obligations irrelevant
C4: va sa du	expr: not PU[A2]/desire for information [Z] evoc: give info [Z]	speaker: commitment to need for info (Z) listener: evaluate + (give info (Z))
A3: ska du ha en tur & retur	expr: info [Z] evoc: accept info[Z]/give info [Y]	speaker: commitment to info (Z) + need for info (Y) listener: evaluate + (give info (Y))
C5: ja/ö	expr: info [Y]; hesitation evoc: continue existing purposes	speaker: commitment to info (Y) listener: evaluate + (continue give info (X))
A4: vilken månad ska du åka	expr: desire for info [W] evoc: give info [W]	speaker: commitment to need info (W) listener: evaluate + (give info (W))

We can see how expectations related to evocative functions and to obligations connected with role and dialog act influence the interpretation of the utterances and the progression of the dialog. The first utterance C1 hup is not a conventional word of Swedish but a sound which, for example, could be used by a solitary speaker as an expression of surprise or fear. In this context, however, given the purpose of the activity and the roles of the interacting parties, it functions as a summons for contact and perception (attention) and a way of initiating the interaction. In utterance A1, it has been assumed that CPU (contact, perception, understanding) as well as acceptance is part of what is being mainly expressed which is why CPU and acceptance have been coded while they have been left out in most other utterances. In C4 `va sa du` (what did you say) has been coded as expressing non-perception/understanding of utterance A2. Turning to "acceptance", we can see that it is left out except in A1 and A2. In C2, thus, acceptance of the task of making a request is implied by the fact that C2 is such a request, and in C5, acceptance of A3 as being a clarification of A2 is implied by the fact that C5 answers the yes/no question contained in A3. Utterances A3 and C5 are both implicit statements expressing beliefs. In the case of A3, it is the belief that A3 is a clarification of A2 and in C5 it is C's belief that he/she wants a return ticket. However, in accordance with what was said above, we code the commitments going with A2 and C5 as commitments to the information. Since the *yes*-answer in C5 is also an implied positive statement that "C wants a return ticket", it carries the default evocative function that A should share this belief (not coded). In A4, A does not object but continues his task which then implies that he, in fact, accepts this belief, i.e., that C wants a return ticket.

In utterance C2, the NP *flyg ti Paris* (flight to Paris), because it is uttered by the customer at the beginning of the activity, can function as a request for information giving rise to an obligation for the agent to furnish that information. The reason this is an obligation rather than just a hoped-for action from the listener is that a positive evaluation on A's part can be expected and that A, by his role is obligated to provide relevant services. Also, since C has entered the role of customer he/she is, in turn, obliged to provide sufficient information for A to do his/her job. Similarly, the requests for specification (in A2, A3 and A4) and clarification (in C4) give rise to obligations to furnish information which are relevant in the activity and motivated by the roles of the two interlocutors.

3.2 The Quarrel between two sisters

For comparison we will now analyze the quarrel in a similar way by first giving an analysis of expressive and evocative functions and then turning to obligations.

Quarrel— Expressive and Evocative Functions and Obligations

Contribution	Expr. and Evocative Function	Obligations Introduced
D1: men herregud <clicking sound>/	expr: irritation evoc: PU/irritation	speaker: commitment to being upset for some reason listener: evaluate + respond
S1: kan du låta bli min freestyle eller	expr: desire for cessation of action/irritation evoc: cessation of action	speaker: commitment to expressed desire listener: evaluate + respond
D2: nä	expr: refusal evoc: PU/irritation	speaker: commitment to refusal listener: evaluate + respond

S2: <a men släng inte ner den nu> <yelling>	expr: desire for cessation of action/irritation evoc: cessation of action	speaker: commitment to expr. desire listener: evaluate + respond
D3: <ingen fara>/den e ändå så gammal <very slowly>	expr: belief evoc: irritation	speaker: commitment to belief listener: evaluate + respond
S3: vadå gammal två dar	expr: protest/info evoc: CPU	speaker: commitment to protest + info listener: evaluate + respond
D4: aa //	expr: CPU evoc: CPU	speaker: commitment to PU listener: evaluate
S4: <sings>	expr: disdain evoc: irritation	no relevant obligation
D5: sluta du e ÄCKLI	expr: desire for cessation of action + belief + irritation evoc: cessation of action	speaker: commitment to desire and belief listener: evaluate + respond

In this dialog , CPU is less taken for granted than in the travel bureau dialog. In utterances D1, D2, S3, and D4, PU or CPU have been included as main evocative functions since getting the other sister to listen and understand seems to be a main evocative intention which can be less taken for granted in a quarrel then in a travel agency dialog. Another difference is that the roles of the two sisters are such that there is no expectation that positive evaluation carries with it an obligation to act. Thus, the only obligation D has after utterances S1 and S2 is to evaluate whether she is willing and able to cease the action S requests not to be done. A third difference is that since utterances D3 and D5 contain explicit statements, we have used the predicate “belief” to code the expressive function and the commitments generated by this. In S3, which is an implicit statement, we have, like in the travel dialog, used the predicate “information”.

4. Discussion

It is fairly clear that the quarrel is different in nature from the travel agency dialog. The conventional expectations associated with the role of teenage sister and the roles of customer and agent are of a different kind. In the travel agency dialog, the roles allow fairly good predictions about what communicative acts it is reasonable to expect and about what the obligations of the two parties are, but this is much more uncertain in the quarrel. The interaction between the sisters is in a sense free of clear role obligations. Instead there is probably a kind of fundamental trust between the two sisters which allows for a breach of some ethical and politeness considerations of obligations as well as for a neglect of obligations generated locally by the communicative acts used by the other party. Thus, D does not placate S by assenting to requests or by trying to lessen the irritation S expresses. Rather she seems to want to tease S, in order to make her more irritated. When S starts to sing, by ignoring D she irritates D. D then answers by insulting S. Ethical considerations involving trying not to hurt the other party are thus diminished and some of the features of what in Conversation Analysis is called “preference organization” don’t seem to be present.

If we consider to what extent the two dialogs exhibit features of cooperation, we see that in the travel agency dialog, the two parties clearly take each other into cognitive consideration. They also cooperate in trying to achieve the common purpose of giving and receiving information about traveling.

In addition, they seem to show each other some ethical consideration. The agent, for example, tells the customer to hold on when he is using his computer to find relevant information. Probably this is also connected with some mutual trust between the parties. Each expects the other party to treat him/her in a way that is correct given her/his roles as customer and agent. This leads to a kind of harmony between communication based on role obligations and communication based on obligations generated by the communicative acts that are used. The travel dialog exhibits what we might call professional cooperation or cooperation strongly influenced by roles in a conventionalized social activity.

Turning to the quarrel, cooperation, if it exists at all, is both less obvious and of a different kind. The two sisters cooperate at least to the extent that they take each other into cognitive consideration. This is shown by the coherence of their interaction. They might also be said to cooperate in the sense that they share the purpose of achieving some kind of mutual understanding. This is shown by the fact that they do seem to interpret each other’s utterances in a reasonable way and respond to them coherently. More controversially, one might also claim that they, after a while, come to share the purpose of irritating each other, which, as we can see, has consequences for how they respond to each other’s utterances. Whether or not quarrelling or mutual irritation can be accepted as a joint purpose depends on whether the resulting interaction merely is the outcome of two individual purposes (where one person wishes to irritate the other person) or whether it has features indicating a joint purpose such as, for example, mutually licensing neglect of various obligations and commitments. The question of whether the two sisters show each other trust and ethical consideration beyond coherent responses, is an even more complicated issue. They are irritating each other and thus being unethical. However, the pain seems to be kept within certain limits. Therefore, it can perhaps be claimed that even though their interaction is not ethically ideal, there is a sense of trust between the two which means that there will be limits to how much the other party can be hurt—a kind of mutual bond of tolerance up to a point. This kind of fundamental trust might be what often differentiates a quarrel between people who are bonded by, for example, siblinghood, marriage, or friendship from a quarrel between strangers or enemies.

In sum, we may therefore conclude that both interactions exhibit cooperation, albeit of different kinds and magnitude. What has been presented so far is an account of some of the main cooperative dimensions of dialog. The goal of a dialog is to allow the participants to share awareness and understanding while at the same time attempting to influence each other. In doing so, the dialog participants often express (and thereby often clarify) their attitudes and emotions.

These goals are realized through communicative actions which are guided by cognitive consideration and often also by different types of ethical consideration and trust. The goals are often further reinforced by being linked to the functional role requirements of a particular activity. The dialog successively progresses as the senders provide expressive and evocative information, which the receiver(s) either explicitly confirm by the use of the feedback system of a language (Allwood 1988 and Allwood, Nivre, and Ahlsén 1992) or implicitly confirm (as being jointly perceived, understood, or accepted) by contributing new information building on the previous contribution.

These cooperative mechanisms of dialog now allow us to explain why there should be such phenomena as “adjacency pairs” (Schegloff and Sacks 1973), “exchange structures” (Sinclair and Coulthard 1975), “dialog grammars” (Moeschler 1989), or “dialog games” (Kowtko and Isard 1991). According to Schegloff and Sacks, adjacency pairs occur as a kind of conventional pairing of one speech act with another, and it belongs to linguistic competence to know how to respond to a given type of speech act. The problem with this approach is that it does not explain what happens when people respond coherently in unexpected ways. Responses such as *shut up* or *why do you say that*, etc., are always possible; the question is why they do not occur very frequently.

The view described above, rather than merely invoking a conventional mechanism, instead suggests that relevant pairings of utterances occur because speakers are cooperative, i.e., to some extent consider each other’s contributions both cognitively and ethically, share purposes, and trust each other.

Thus, in evaluating another person’s contribution it would not be cooperative to just ignore it or to reject it out of hand without reason. Instead, we usually try to at least perceive, understand, continue, and, if we have no reasons against it, comply with the main evocative intention. When such compliance occurs a successful “adjacency pair” is produced. However, what has occurred is not merely an instance of a conventional mechanism but rather a voluntary ethically motivated action.

The regular and expectable features of dialog should be seen as an outcome of cooperation in which expressive and evocative features of contributions, on the basis of obligations, are evaluated and responded to by new contributions with new expressive and evocative features. In this process, a large part of the bond and coherence between utterances is provided by meeting the obligations given by general ethics, activity roles and particular communicative acts. Since it has further been claimed that cooperation is a matter of degree, which is based on the willingness and ability of the participants, regular dialog features can, at any moment, be modified, changed, or interrupted. The fact that this does not happen more often than it does is a sign of the strength of the role that cooperation plays in human social life.

References

Allwood, J. (1976) Linguistic Communication as Action and Cooperation. *Gothenburg Monographs in Linguistics 2*. Göteborg University, Department of Linguistics.

Allwood, J. (1977) A Critical Look at Speech Act Theory. In Dahl (Ed.). *Logic, Pragmatics and Grammar*, Lund, Studentlitteratur, pp. 53-69.

Allwood, J. (1978) On the Analysis of Communicative Action. In Brenner (Ed.) *The Structure of Action*. Oxford: Basil Blackwell.

Allwood, J. (1987) A Semantic Analysis of Understanding. In V. Rosén (Ed.) *Papers from the Tenth Scandinvian Conference of Lingustics,* University of Bergen, Department of Linguistics and Phonetics.

Allwood, J. (ed) (1988) Feedback in Adult Language Acquisition (Final Report II). *Ecology of Adult Language Acquisition* (ESF).

Allwood, J. (1994) “Obligations and Options in Dialogue”. In *Think* , vol 3, May 1994, ITK, Tilburg University.

Allwood, J. (1995) An Activity-based Approach to Pragmatics. *Gothenburg Papers in Theoretical Linguistics, 76*. Göteborg University, Department of Linguistics. Forthcoming in Bunt and Black (Eds.) *Approaches to Pragmatics*.

Allwood, J., Nivre, J. and Ahlsén, E. (1990) “Speech Management: On the Non-Written Life of Speech”. *Gothenburg Papers in Theoretical Linguistics, 58*. University of Göteborg, Department of Linguistics. Also in *Nordic Journal of Linguistics*, 13.

Allwood, J., Nivre, J. and Ahlsén, E. (1992) On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics* 9, 1–26.

Austin, J. A. (1962) *How to Do Things with Words*. Harvard University Press.

Bühler, K. (1934) *Sprachtheorie*. Jena: Fischer.

Grice, H. P. (1975) Logic and conversation. In P. Cole and J. L. Morgan (Eds.) *Syntax and Semantics Vol. 3: Speech Acts*. New York: Seminar Press, 41–58.

Kowtko, J. and Isard, S. (1991) Conversational games within dialog. *Proceedings of the ESPRIT Workshop on discourse Coherence*, University of Edinburgh, 4-6 April, 1991.

Moeschler, J. (1989) *Modélisation du dialog*. Paris: Hermès.

Sadek, M. D. (1991) Dialogue acts are rational plans. In *Proceedings of the ESCA/ETR workshop on multi-modal dialogue*.

Schegloff, E. and Sacks, H. (1973) Opening up closings. *Semiotica*, 7.4, 28-327.

Sinclair, J. and Coulthard, R.M. (1975) *Towards and Analysis of Disourse. The English Used by Teachers and Pupils*. London: Oxford University Press.

PART II

COMMUNICATION MANAGEMENT-RELATED CODING

FEEDBACK

Coding Manual

Joakim Nivre, Jens Allwood, Elisabeth Ahlsén, Maria Björnberg, and Alexandra Weilenmann

Version 1, 1999

1. Introduction

The purpose of this document is to formulate principles for the coding of *feedback*.

2. Transcription

The first step in the preparation of data for coding is to produce a transcription which is segmented into words and utterances, and where overlapping speech is consistently and unambiguously marked. The notion of utterance is defined in the following way:

Definition 1 An utterance by a speaker α is a stretch of speech produced by α, bounded by silence or by the speech of another speaker.[2]

3. Feedback

3.1 Feedback Units

The first step in the coding consists in identifying feedback units (FBUs) according to the following definition (cf. Allwood 1988a, 1988b):

Definition 2 A feedback unit is a maximal continuous stretch of utterance (occuring on its own or as part of a larger utterance), the primary function of which is to give and/or elicit feedback.

[2] Note that in order to allow for pauses within an utterance, a distinction must be made between *silence* (which does not belong to an utterance) and *pause* (which is considered to be part of an utterance). For the moment, we have no strict operationalization of this distinction to offer. Note also that a pause within an utterance counts as such only if it is not filled by the speech of another speaker. In the latter case, it counts instead as an utterance boundary. We are aware that this leads to a certain arbitrariness in the segmentation of utterances, but we nevertheless feel that this is the best (semi-formal) operationalization that can be achieved at present.

Let us consider a few examples:3

(1) A: kommer du
B: ja
A: kan du [1 ta me en]1 penna
B: [1 va sa du]1
B: okej // vill du ha en egen
A: ja de vill ja

In this (invented) dialog, we can distinguish four FBUs: `ja`, `va sa du`, `okej`, and `ja de vill ja`. The first and third of these consist of a single word, while the other two are larger phrases. The third one (`okej`) is part of a larger utterance, while the other three constitute utterances by themselves.

3.2 Structure

After the identification of FBUs, we proceed to a structural classification of these units. First of all, the units are coded with respect to grammatical categories. For one-word units, this means assigning one of the lexical categories in Table 1 to the word in question.

Lexical Category	**Code**
Feedback word	`fb`
Interjection	`interj`
Noun	`noun`
Adjective	`adj`
Verb	`verb`
Preposition	`prep`
Adverb	`adv`
Proper name	`pn`
Pronoun	`pron`
Conjunction	`conj`
Complementizer	`comp`
Determiner	`det`
Auxiliary	`aux`

Table 1: Lexical Categories

The first category (feedback word) corresponds to the category of primary feedback words in Allwood (1988a) and is exemplified by words such as `ja`, `nej`, `mm`, etc. The remaining lexical categories (when used for feedback) are collectively referred to as secondary feedback words in Allwood (1988a). In example (1), the units `ja` and `okej` are both assigned the lexical category FB, while words such as `precis` and `bra` would be coded as adverbs (ADV).

[3] In this document, we use non-disambiguated speech in the examples

In feedback units consisting of more than one word, each immediate constituent is assigned a grammatical category, either a lexical one (see above) or one of the syntactical categories in Table 2.

Syntactical Categories	**Code**
Sentence	s
Noun phrase	np
Verb phrase	vp
Adjective phrase	ap
Adverb phrase	advp
Prepositional phrase	pp
Other	other

Table 2: Syntactical Categories

Thus, in example (1), the unit va sa du is assigned the syntactical category S, while the unit ja de vill ja is segmented into ja, which has the lexical category FB, and de vill ja, which has the syntactical category S. A unit such as ja precis is segmented into ja (FB) and precis (ADV).4

In addition to the coding of grammatical categories, the structural classification also involves coding for structural operations, which may be subdivided into phonological, morphological, and contextual operations. If several feedback words occur together and fall into distinct groups, these should be distinguished. For example, the sentence ja just de ja should be analyzed as ja (fb), just de (advp), and ja (fb).

Phonological operations

1. **Lengthening** is an operation that can be applied to any word or phrase.
   Example: ja => ja:
2. **Continuant reduplication** is an operation that is mainly used with primary feedback words. It comes in three varieties:
   (a) pure, e.g., ja => jaa
   (b) with glottal stop, e.g., ja => ja’a
   (c) with glottal fricative, e.g., ja => jaha
3. **Vowel addition** is mainly used with primary feedback units.
   ja => ja
4. **Truncation** is mainly used with primary feedback units.
   ja => a
5. **Ingressive** is mainly used with primary feedback words.
6. **Prosodic modification** (other than lengthening) occurs with all kinds of words and phrases.

[4] Major syntactic phrases such as noun phrases and sentences can in principle be further analyzed into lexical categories, but in most cases the syntactic categorization will be sufficient. As for higher level combinations, such as `feedback word + sentence’ or `feedback word + adverb’, there is no need to code the category combinations separately, since these codes can be derived automatically from the coding of the constituents

Morphological operations

1. **Reduplication** occurs mainly with primary feedback words.
   `ja` => `jaja`
2. **Inflection/derivation**
   `hej` => `hejsan`
3. **Compounding**
   `ja` + `då` => `jadå`
4. **Reduction**
   `jaha` => `ha`

Contextual operations

1. **Repetition** (verbatim) of some part of the immediately preceding utterance.
   A: `har du en penna`
   B: `penna`
2. **Reformulation**, i.e., reformulation by means of deictic and anaphoric expressions of some part of the immediately preceding utterance.
   A: `har du en penna`
   B: `ja de har ja`

Tags	**Values**

`phon_op`	`lengthening`
	`cont_redupl (pure)`
	`cont_redupl (fricative)`
	`cont_redupl (stop)`
	`vowel_addition`
	`truncation (pure)`
	`ingressive`
	`prosody`

`morph_op`	`reduplication`
	`derivation`
	`compounding`
	`reduction`

`context_op`	`repetition` `reformulation`

Table 3: Tags for Structural Operations

Note that it is often necessary to assign several codes to the same segment. For example, in a feedback unit such as `jaa de gör ja`, the segment `jaa` is assigned the codes `lexcat=fb` and `phon_op=cont_redupl(pure)`, while the segment `de gör ja` is coded `syncat=s` and `context_op=reformulation`.

3.3 Position and Status

The coding of *position and status* concerns the status and position of an entire FBU within a larger utterance. Thus, this coding indicates whether the FBU is

single (i.e., constitutes an entire utterance by itself),
initial in an utterance,
medial in an utterance, or
final in an utterance.

The first coding, *single*, thus indicates the status whether the FBU is an utterance in its own right or not. The final three codings indicate the position for an FBU which is not single within a larger utterance. In example (1), the first two FBUs (`ja` and `va sa du`), as well as the last one (`ja de vill ja`), are *single*, while the third one (`okej`) is *initial*.

3.4 Function

Two aspects of the *function* of FBUs are coded:

1. Function type
2. Attitudes

The latter aspect applies mainly to *feedback givers* (see below).

3.4.1 Function type

By *function type* is meant a broad classification of feedback functions into:

giving feedback (`give`),
eliciting feedback (`elic`), and
giving *and* eliciting feedback (`give_elic`).

In example (1), `ja`, `okej,` and `ja de vill ja` are givers, while `va sa du` is a giver–elicitor.

3.4.2 Attitudes

The coding of attitudes can be broken down in three parts:

CPU attitudes
Acceptance attitudes
Other attitudes

The term *CPU attitudes* is used to refer to the attitudes of *contact* (cont), *perception* (perc), and *understanding* (und) (cf. Allwood 1988a). These basic communicative functions are normally coded only when "marked", i.e., when they are *negative* and/or *explicit* (as opposed to the "unmarked" case which is *positive* and *implicit*). In our example, only the unit va sa du needs to be coded for negative perception (perc(neg)) and understanding (und(neg)). In general, *CPU attitudes* is coded as positive, negative, or not relevant/applicable.

By *acceptance attitudes* we mean the attitudes of acceptance or non-acceptance, occurring especially after communicative acts such as statements, questions, requests, and offers (cf. Allwood, Nivre, & Ahlsén 1992). Primarily, these attitudes concern the main evocative intention of communicative acts, i.e.,accepting a statement as correct and worthy of belief, accepting in order to answer a question, or accepting in order to carry out a request. Secondarily, the attitudes can concern acceptance of a turn and/or a communicative act. Since these two secondary functions closely correspond to the CPU attitudes in such a way that *acceptance of turn* corresponds to *contact* and *acceptance of communicative act* corresponds to *perception* or *understanding* of communicative act, they are not separately coded.

Acceptance attitudes can be coded as positive (acceptance), negative (non-acceptance), or not relevant/applicable. Finally, expressive features which go beyond CPU attitudes should be coded. Examples of such expressive features are epistemic, emotional, and discursive attitudes, as well as some speech act-like functions. The following list gives some examples: hesitation, uncertainty, surprise, reminder, anger, happiness, sadness, contempt, friendliness, irony, support, polite, concession, admission, objection. No Label means that no specific expressive feature has been noted, i.e., that the attitude is neutral or too difficult to decide on. No attempt is made to clearly distinguish attitude labels from speech act labels since many terms can be used in both senses, e.g., *support* and *object.*

Tag	**Value**
cont	pos neg
perc	pos neg
und	pos neg
acc_evoc_function	pos neg
other_expr_features	surprise anger happiness sadness contempt prosodic features

Table 4: Tags for attitudes

3.5 Some problematic cases

3.5.1 Primary feedback words

The following Swedish words count as primary feedback words:

ja
jo
nej
nä
nja
m
okej
va

Nja should also be regarded as a primary feedback word, often used when the speaker is doubtful about the content of the previous utterance. In such a case, acceptance should be marked as negative and other expressive features: doubt should also be coded. Jaha is counted as a primary derived-feedback word.

3.5.2 Other feedback words

For all the words, the other expressive features that have been suggested are default values, which can be invalidated by context.

naä occurs in contexts where the speaker is sceptical and does not really agree with the previous speaker. For example: naä de menar ja inte. Naä should be coded with acc_evoc_function negative
other expressive feature: hesitation, truncation, and vowel addition.

naäj see naä above

ne variant of nä and should be coded the same way

na acceptance: negative
other expressive feature: hesitation

tja should not be coded as negative or positive concerning content
other expressive feature: hesitation should be marked

3.5.3 Pauses

Whether pauses should be regarded as part of an utterance or not is to be decided on the basis of the context. This is relevant in cases where feedback utterances end with a pause, as in the following example:

A: mm //

Here we have decided not to consider the pause as part of the utterance when coding the position and status of the feedback segment. `mm` will be coded as `position:single`.

3.5.4 Hesitation sounds

When coding feedback, don’t pay any notice to hesitation sounds like *e1*, etc:

A: just e1 precis

This entire phrase should just be coded as an `advp`.

3.5.5 Words and phrases of Greeting

Hej and **Hej då**

`Hej` should be coded as feedback. This word is regarded as feedback giving as well as eliciting, depending on the situation. If there are two persons greeting each other, the first `hej` should be coded as `elicit`, and the response-`hej` as `give`. `Hej` belongs to the lexical category `Interjektion` and `Hej då` should be coded as `Interjp`. Other than that, `hej då` should be coded like `hej`.

Välkommen should be considered as *feedback-eliciting*, and should thus be coded as `elicit`. The word is an *interjection*.

Var så god should be coded as `Interjp`.

References

Allwood, J. (ed) (1988a) Feedback in Adult Language Acquisition (Final Report II). *Ecology of Adult Language Acquisition* (ESF).

Allwood, J. (1988b) “Om det svenska systemet för språklig återkoppling”. In Linell, P., Adelswärd, V., Nilsson, T. and Pettersson, P. A. (eds) *Svenskans beskrivning* 16. SIC 21a. University of Linköping: Department of Communication Studies.

Allwood, J., Nivre, J. and Ahlsén, E. (1992) On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics* 9, 1–26.

ADDRESSEE,

TURN AND SEQUENCE MANAGEMENT

Coding Manual v2.0

Jens Allwood and Maria Björnberg
January 2000

1. Categories

This manual contains coding for four different categories:

- Addressee
- Opening of Activity
- Closing of Activity
- Overlap function

It can be noted that the last three categories are partly overlapping in the sense that the same phenomena may be coded in more than one category, though from different perspectives.

1.1 Addressee

The addressee of an utterance can be coded using the following seven categories:

- Addressing an unidentified participant. (It's not clear who the speaker is addressing.)
- Addressing all participants in the conversation (explicitly, implicitly)
- Addressing a specific person (explicitly, implicitly)
- Addressing a group of persons (explicitly, implicitly)
- Addressing nobody
- Addressing oneself
- Addressing other
  - massmedial
  - eavesdropper

Some of the categories, as seen, are divided into *explicit* and *implicit* addressing. *Explicit addressing* is pointing out the addressee(s), for example by addressing him/her by name. If the addressee(s) is/are understood without being mentioned, the addressing is *implicit.*

The category *addressing other* refers to the addressing of people who are not participating in the conversation. The category is divided into massmedial and eavesdropper. An example of massmedial addressing is addressing the listeners of a radio or TV program. Eavesdroppers could be involved (they are not addressed, but are persons who are listening without being addressed) for example at a travel agency, where an eavesdropper might be listening while the travel agent is talking to another customer.

1.2 Opening of Activity

Coding *opening of activity* consists of coding the utterance(s) that open the activity. The example below is from a conversation taking part when a customer walks into a travel agency and asks for help. These four utterances should all be coded as *opening of activity*.

Example (R and S are speakers):

R: hej
S: hej
R: / ska vi hjälpa dej här
S: m

1.3 Closing of Activity

Coding *closing of activity* consists of coding the utterance(s) that close the activity. The example below is the end of a conversation at a travel agency. These four utterances are coded as *closing of activity*:

Example:

P: okej a men+
J: du få fundera lite du e0 välkommen igen
P: a:
J: mm

1.4 Overlap function

Coding *overlap* presupposes dividing utterances into smaller parts, since very often a part of an utterance is overlapped. Overlapping utterances/parts of utterances are marked in the transcription with square brackets. Instances of overlap are coded with the following two functional categories:

- Interruption. Here we distinguish the interrupted utterance from the interruption.
- Other function. Overlaps can occur for many reasons other than being part of an interruption. Some common functions, which may occur separately from or simultaneously with an interruption, are the following (other functions may be added):
    - *Giving attention*
    - *Affirmation*
    - *Acceptance*
    - *Reaffirmation*
    - *Reminder*
    - *Excuse*
    - *Continuation*
    - *Hesitation*
    - *Disagreement*
    - *Lack of hearing or understanding*
    - *Other*

Conversation	**Overlap coding**
P: mm å m a precis då e kan man åka / [14 nä]14 som helst	Interrupted utterance
J: [14 fölåt]14	Interruption Overlap function: Lack of hearing or understanding

OWN COMMUNICATION MANAGEMENT

Coding Manual v1.0

Jens Allwood, Elisabeth Ahlsen, Joakim Nivre, Staffan Larsson

October 10, 1997 (revised January 2001)

1. Introduction

The purpose of this document is to formulate principles for the coding of *own communication management* (OCM). OCM is an umbrella term for the processes speakers use to manage their own linguistic contributions to communicative interaction (e.g., planning phenomena, repair, editing, self-correction, etc.). We can distinguish between two descriptive perspectives for these processes: one related to the function of the processes, and one based on the expressions used in the structure of the processes. However, the two perspectives have some overlap so that the perspective which we will be calling “expression structure” is not exclusively oriented to linguistic surface structure but is rather primarily focused on linguistic structure, likewise what we are calling “function” relies on linguistic structures but is primarily focused on function.

The examples in this coding manual are reproduced according to the transcription standard specified in Nivre (1999a, 1999b).

2. Computer Tools

Coding is simplified considerably by computerized transcription coding tools. One such tool is TRACTOR (Transcription Coding Tool). This tool and its use are described in a separate document (Larsson 1997). The present manual is in large part independent of how the practical process of coding takes place. The TRACTOR version of the OCM coding schema makes use of coding labels in English.

3. Function Coding

As mentioned earlier, OCM phenomena can be classified both from a functional perspective and with respect to utterance structure. We can distinguish between two types of OCM function:

1. **Choice-related OCM** helps the speaker to gain time for processes having to do with continual choice of content and types of structured expressions, especially memory searches, hesitation, and planning.

Example: de e en **va heter de** valkyria
*it's a **whaddya call it** valkyrie*

2. **Change-related OCM** helps the speaker (on the basis of various internal and external feedback processes) change already-produced content, structure, or utterances. Examples of change-related OCM are self-repair and self-correction.

Example: de e en blå **ja1 menar** röd bil
*it's a blue **i mean** red car*

These functions are coded according to the following procedure:

1. Mark off the maximal[5] sequence of words which are used for choice- or change-related OCM. (This sequence corresponds to the text in boldface in the examples above).
2. Choose the appropriate code.

Note that it can sometimes be difficult to determine whether a segment should be marked as "change" or "choice", e.g., when a speaker at first appears to be searching for a word (choice), but then gives up and instead begins thinking of a way to modify that which has already been said (change). In such a case, the coder can only rely on intuitions and code the first segment as change, the second as "choice".

In other cases it can be motivated to code one and the same segment as both change and choice, especially when one and the same segment contains more than one OCM structure (see section 4).

Example: ja vill ha banan [[nä:]$_{\text{simple\_ocm\_expr}}$]$_{\text{lengthening}}$ jordgubbsglass[6]
I want banana [[no:]$_{\textit{simple\_ocm\_expr}}$ *]*$_{\textit{lengthening}}$ *strawberry ice cream.*

In this case "nä" should be coded as both choice and change, since "nä" signals that the speaker wants to modify what has already been said, while the lengthening of "ä" indicates that the speaker wants to buy time for choosing how to continue. Note that coding under both functional categories should not be used as a way to indicate uncertainty as to which OCM function a segment has. In dubious cases the coder must decide which function seems to be the dominant one and code under this function only.

[5] For the sequence to be maximal requires that, instead of coding a coherent sequence with multiple tokens of one and the same OCM function, the entire sequence should be coded as one token of the OCM function.
Example: de e en [ä0]$_{\text{choice}}$ [//]$_{\text{choice}}$ blå bil ___ de e en [ä0 //]$_{\text{choice}}$ blå bil
it's a [ä0]$_{\textit{choice}}$ *[//]*$_{\textit{choice}}$ *blue car ___ it's a [ä0 //]*$_{\textit{choice}}$ *blue car*

[6] This example is invented.

4. Expression Structure

A number of different expressions and operations can be used to realize OCM functions. Many of these structures can also be used for other purposes, but where OCM coding is concerned, we are only interested in OCM-related occurrences.

Among the expression structures which realize OCM, we can distinguish between basic and complex OCM structures, where the latter are combinations of the former.

4.1 Basic OCM Structures (basic_ocm_features)

Basic OCM Expressions (basic_ocm_expressions)

1.Pause, i.e., absence of speech and gestures during a turn. Note that only OCM pauses, i.e., pauses with OCM function, are to be coded (pauses are counted here as a kind of expression, by stipulation).

> **Example:** de betyder att [//]$_{\text{pause}}$ alla försöker va ett steg före hela tiden
> *it means that [//]*$_{\textit{pause}}$ *everyone is trying to be one step ahead all the time*

2. Simple OCM expressions (simple_ocm_expression), e.g., eh, uh, m, liksom *(like)*, eller *(or)*, nä *(no)*.

> **Example:** ja kom å tänka på [äh]$_{\text{simple\_ocm\_expr}}$ torpet
> *i got to thinking about [uh]*$_{\textit{simple\_ocm\_expr}}$ *the cottage*

3. Explicit OCM phrases (explicit_ocm_phrase), e.g., vad heter det *(whaddya call it)*, rättare sagt *(more precisely)*, så att säga *(so to speak).*

4. Other OCM sounds (other_ocm_sound) which are difficult to classify, e.g., smacking, sighing, etc.

Basic OCM Operations (basic_ocm_operations)

1. Lengthening of continuants (lengthening_of_continuant), i.e., of sounds that can be sustained.

> **Example:** ja hade ju hoppat över dom här konstiga figurerna fö ja [inte:]$_{\text{lengthening of continuant}}$ ja0 alltså
> *course i had skipped over these strange figures because i [didn:]*$_{\textit{lengthening of continuant}}$ *yeah anyway*

2. Self-interruption (self_interruption), i.e., a speaker interrupts himself in the middle of a word or phrase. Self-interruption can thus occur in the middle of a word, usually marked with + in the transcription, or between two words. The latter case is not indicated explicitly in the transcript, but usually only appears as a sudden interruption of the preceding syntactic structure, possibly

followed by simple OCM expressions, e.g., pauses or hesitation expressions. When coding self-interruption between two words, mark the word that occurs immediately before the interruption.

Example: så [ma+]$_{\text{self\_interruption}}$ a just de a å
so [ma+]$_{\textit{self\_interruption}}$ *yeah exactly yeah and*

Example: men [vi]$_{\text{self\_interruption}}$ ja1 tänkte att vi får väl lösa de0
but [we]$_{\textit{self\_interruption}}$ *i was thinking that we'll just have to solve it*

3. Self-repetition (self_repetition), i.e., the speaker repeats something he/she has just said. That which is repeated can be either a word or an entire phrase, and OCM expressions can sometimes appear between the repetitions. Note that if anything other than OCM expressions occurs between the repetitions, then it does not count as self-repetition.
This definition can be captured in a simple schema for self-repetition:

Schema SR: A (OCM) A

This schema is interpreted such that a word or phrase A followed by the same word or phrase A, possibly with (basic) OCM structures intervening, constitutes a case of repetition[7].

Example: de e valt bara **bara** för att de ska
it is chosen just ***just*** *so it will*

Example[8]: de e valt bara för **[bara för]**$_{\text{self\_repetition}}$ att de ska
it is chosen just so ***[just so]***$_{\textit{self\_repetition}}$ *it will*

Example[9]: de e valt bara ä / **[bara]**$_{\text{self\_repetition}}$ för att de ska
it is chosen just uh / ***[just]***$_{\textit{self\_repetition}}$ *so it will*

These expression structures are coded according to the following procedure:

1. Mark off a basic OCM expression, or an expression (word) which realizes a basic OCM operation (in the example above, this sequence corresponds to the text in boldface). Note that in the case of self-repetition, only the repetition(s) should be marked.

2. Choose the code whose definition fits the marked segment.

These features can occur in isolation or in combination. When they are combined, they can be applied to the same segmental expression, e.g., lengthening of a vowel in a simple OCM expression, which in itself expresses a choice function (eh e:h). They can also occur in succession, e.g., a pause filled by a simple OCM expression (// eh).

[7] The two occurrences of A do not have to be completely identical; they can for example be different phonetic realizations of the same word/expression, or one occurrence may be interrupted. The same goes for the schemata given in Chapter 4.2.
[8] This example is invented for pedagogical purposes.
[9] This example is invented for pedagogical purposes.

Example: för att inte [ääh]$_{\text{simple\_ocm\_expression}}$ [//]$_{\text{pause}}$ [eh]$_{\text{simple\_ocm\_expr}}$ för att hålla en del gröder vid liv
in order to not [uhh] $_{\text{simple\_ocm\_expression}}$ *[//]*$_{\text{pause}}$ *[uh]*$_{\text{simple\_ocm\_expr}}$ *in order to keep some of the crops alive*

4.2 Complex OCM Operations (complex_ocm_operations)

Complex OCM operations[10] is an umbrella term for different ways of modifying the linguistic structure. All complex OCM operations thus have a change function and therefore do not need to be coded for function. Complex OCM operations always involve a self-interruption, which is often complemented by a number of other basic OCM structures.

Schematic definitions of the complex OCM operations are given below. These definitions are not absolute; that is, they do not always have to be strictly adhered to. However, the coder should always make a note of those cases where an utterance segment is coded as realizing a certain OCM operation even though it does not match the schematic definition. Ideally the entire utterance should be written down, as well as the name of the current transcription. When the coding is being done with TRACTOR, the numbers which designate the position of the utterance in the transcription should also be indicated. These cases should then be discussed with the rest of the coding team, since in some cases they can motivate revisions of the coding manual.

Note that (self-)interruption is represented by + in the schematic definitions. This does not imply that these interruptions must always be indicated by a + in the transcription; sometimes the interruption can occur after an entire word has been pronounced, and then the interruption may appear as nothing more than a pause.

The symbol (OCM) also appears in the definitions. This means that (basic) OCM expressions can sometimes occur here. In general, OCM expressions can always occur after self-interruption.

The abbreviations LC and RC can be read as “Left Context” and “Right Context”, respectively. The numerals 1 and 2 indicate whether a schematic element is appearing in its first or second instantiation. The second instantiation (e.g., LC2) thus constitutes a repetition of the first (in this case LC1).

Certain schematic elements appear within parentheses. The parentheses indicate that the element is optional, i.e., that it can, but does not have to, occur. Note however, that if one instantiation of a schematic element (e.g., RC1) is included, then the other instantiation (in this case RC2) must also be included.

In order to increase the readability of the examples below, all functional and structural coding except for the described operation has been omitted.

[10] Since there are no complex OCM expressions, the category “complex OCM expressions” is subsumed under “complex OCM operations”.

1. **Deletion** occurs when material is clearly omitted from the repetition.

Schema D: LC1 deleted (RC1) + (OCM) LC2 (RC2)

> **Example:** men [de e]$_{LC1}$ [ju]$_{deleted}$ [farlit]$_{RC1}$ // [de e]$_{LC2}$ [farlit]$_{RC2}$ för naturen
> *but [it is]*$_{LC1}$ *[of course]*$_{deleted}$ *[bad]*$_{RC1}$ *// [it is]*$_{LC2}$ *[bad]*$_{RC2}$ *for the environment*

> **Example:** men [de e]$_{LC1}$ [ju]$_{deleted}$ // [de e]$_{LC2}$ farlit för naturen[11]
> *but [it is]*$_{LC1}$ *[of course]*$_{deleted}$ *// [it is]*$_{LC2}$ *bad for the environment*

2. **Insertion** occurs when material is clearly inserted into (the middle of) the repetition.

Schema I: (LC1) RC1 + (OCM) (LC2) inserted RC2

> **Example:** De måste han va för att han [snabbt]$_{LC1}$ [på+]$_{RC1}$ [snabbt]$_{LC2}$ [ska kunna ta beslut]$_{inserted}$ [på]$_{RC2}$ en tiondels sekund
> *He has to be in order to [quickly]* $_{LC1}$ *[in+]*$_{RC1}$ *[quickly]*$_{LC2}$ *[be able to make a decision]*$_{inserted}$ *[in]*$_{RC2}$ *a tenth of a second.*

Insertion can also occur without resumption[12].

> **Example:** men [vi0]$_{RC1}$ [ja1 tänkte att]$_{inserted}$ [vi0]$_{RC2}$ får väl lösa de0[13]
> *but [we]*$_{RC1}$ *[i was thinking that]* $_{inserted}$ *[we]*$_{RC2}$ *will just have to solve it*

3. **Substitution** occurs when material is clearly replaced in the repetition.

Schema S: (LC1) substituted (RC1) + (OCM) (LC2) substitute (RC2)

> **Example:** så de [känns]$_{LC1}$ [som]$_{substituted}$ [de e va+]$_{RC1}$ [känns]$_{LC2}$ [att]$_{substitute}$ [de e valt]$_{RC2}$ bara för att
> *so it [seems]*$_{LC1}$ *[like]*$_{substituted}$ *[it is cho+]*$_{RC1}$ *[seems]*$_{LC2}$ *[that]*$_{substitute}$ *[it is chosen]*$_{RC2}$ *just so*

Substitution, too, can occur without resumption if the substituted element (substituted) and the substituting element (substitute) have the same role in the sentence.

> **Example:** de [blir]$_{substituted}$ [väldit]$_{RC1}$ [låter]$_{substitute}$ [väldit]$_{RC2}$ jobbit
> *it [gets]*$_{substituted}$ *[really]*$_{RC1}$ *[sounds]*$_{substitute}$ *[really]*$_{RC2}$ *annoying*

[11] This example is invented for pedagogical purposes.

[12] When a part of that which was said before a self-interruption is repeated after the interruption, it is referred to as *resumption* (Swe. återknytande).

[13] Note that this could also be considered a case of substitution rather than insertion, where ja1 replaces vi0. To determine which operation is in evidence, the coder must use his own linguistic intuitions. In such cases it is often very helpful to listen to the sound recording.

Sometimes substitution occurs with the simplest structure imaginable, where even the element RC in the schema above is omitted. Here too, the substituted element and the substituting one must have the same role in the sentence.

> **Example:** ska vi återkalla lite grann [om]$_{\text{substituted}}$ / [AV]$_{\text{substitute}}$ de vi talade om förra veckan
> *can we go back for a moment [about]$_{\text{substituted}}$ / [TO]$_{\text{substitute}}$ what we talked about last week*
>
> **Example:** han talade om [våra]$_{\text{substituted}}$ e1 eller [vår]$_{\text{substitute}}$ nya datoriserade värld
> *he talked about [ours]$_{\text{substituted}}$ uh or [our]$_{\text{substitute}}$ new computerized world*

4. **Reordering** occurs when material is clearly reordered in the repetition.

Schema R: LC1 re_a re_b RC1 + (OCM) LC2 re_b re_a RC2

> **Example:** men sen [hade]$_{\text{LC1}}$ [ja]$_{\text{re\_a}}$ [inte]$_{\text{re\_b}}$ [lä+]$_{\text{RC1}}$ [hade]$_{\text{LC2}}$ [inte]$_{\text{re\_b}}$ [ja]$_{\text{re\_a}}$ [läst]$_{\text{RC2}}$ dom siderna
> *[which]$_{LC1}$ [at that time]$_{re\_a}$ [i hadn't re+]$_{re\_b}$ [which]$_{LC2}$ [i hadn't read]$_{re\_b}$ [at that time]$_{re\_a}$*[14]

Assuming you have found an utterance segment which matches one of the above definitions, do the following to code the function:

1. Mark the relevant part (word or phrase) of the segment.

2. Determine which code corresponds to the structure type and the schematic part the segment plays in the definition of that structure, e.g., *inserted* or *RC1*.

[14] This example was fabricated to serve as an approximation of the Swedish reordering example.

References

Larsson, S. (1997): *TRACTOR Användarmanual*. Department of Linguistics, Göteborg University.

Nivre, J. (1999a): *Modifierad Standardortografi Version 6*. Department of Linguistics, Göteborg University.

Nivre, J. (1999b): *Transcription Standard Version 6.2*. Department of Linguistics, Göteborg University.

PART III

GRAMMATICAL CODING

PARTS OF SPEECH CODING

One of the ways of coding grammatical structure is an automatic, probabilistic coding of parts of speech. This coding scheme contains the following categories:

Tag	**Part of Speech**
adj	Adjective
adv	Adverb
art	Article
conj	Conjunction
fb	Feedback word[15]
inf	Infinitive marker
interj	Interjection
n	Noun
num	Numeral
ocm	OCM word[16]
part	Particles
pron	Pronoun
v	Verb

[15] The part of speech “feedback words” includes primary feedback words like “ja”, “jo”, “nej”, “nä”, “nja”, “m”, “okej”, and “va”.

[16] OCM (Own Communication Management) words are certain words that always or often have OCM function, for example hesitation sounds like “eh” and “m”.

MAXIMAL GRAMMATICAL UNITS

Coding Manual

Jens Allwood, Maria Björnberg, and Alexandra Weilenmann
Version 1, April 19, 1999

1. Principles for coding maximal grammatical units

In coding maximal grammatical units, one should code above all according to the principle of trying to find the largest units possible; that is, mainly complete sentences. These are coded according to the schema “sentences.pl”. In spoken language, many expressions occur which are not sentences, in which case one should try to find complete phrases, and code them according to the schema “phrases.pl”. As a third resort, if it is not possible to find sentences or phrases, one should code each individual word with the appropriate word-class. This is done according to the schema “parts_of_speech.pl”. Each of these schemata contains different categories, and this manual explains what should be coded under which category.

2. The coding schema *sentences.pl*

The coding schema sentences.pl contains the following categories:

declarative_s/1
disj_question/1
exclamative_s/1
imperative_s/1
wh_question/1
yes/no_question/1

All complete sentences are coded according to this schema. If the sentence contains pauses, hesitation sounds, or repetitions and the like, one should ignore these and code it as a sentence regardless. OCM-phenomena are separately marked and coded above the sentence. Indirect speech also counts as belonging to the sentence. “He said that he is coming” thus counts as one sentence.

Declarative sentences are coded as declarative_s. Three different types of questions can be coded: disj_question (disjunctive questions), wh_question (questions which begin with a wh-phrase), and yes/no_question (questions which can be answered with yes or no). Exclamatory and imperative sentences can also be coded.

3. The coding schema *phrases.pl*

3.1 Coding schema

The coding schema phrases.pl contains the following categories:

adjp/1
advp/1
conj/1
fbp/1
np/1
nump/1
pp/1
subordinate_clause/1
vp/1

3.2 Adjp

Adjp stands for adjective phrase, and it is thus adjective phrases which are coded here.

3.3 Advp

Advp stands for adverb phrase. This includes adverb phrases, but also adverbials of different kinds, as long as they cannot be counted as prepositional phrases. An example:

$P: ja:0 <1typ den: ä1: tredje fjärde april>1 / <2nån gång där>2
$P: yes: <1 like the: uh: third fourth of April>1 / <2 some time in there>2

Here both <1> and <2> will be coded as advp.

3.4 Conj

Conjunction phrases are coded as conj, e.g., “in order to”, “so that”.

3.5 Fbp

Fbp stands for feedback phrase. For something to be considered (and coded) as a feedback phrase in MaxGram, the phrase must contain a primary feedback word. Primary feedback words include:

ja	*yes*
jo	*yes indeed*
nej	*no*
nä	*neah*

nja	*nyeah*
m	*m*
okej	*okay*
va	*what*

This then entails that “just det” *(exactly)* and “just precis” *(precisely)* are not to be coded as feedback phrases. These two should instead be coded as adverb phrases. When a feedback word is combined with other words, as in “yes, exactly, indeed” or “yeah sure”, everything can be coded as fbp, depending on whether it seems to have been said coherently or not. For example, if there is a pause in the middle, then the words should be coded individually instead.

3.6 Np

Noun phrases are coded as np. Note that this refers to noun phrases consisting of several words! A pronoun or a proper name can constitute a noun phrase in itself, but these are to be coded as pronoun or noun, respectively, under the parts_of_speech schema. Complex noun phrases which are adverbial are coded as advp.

3.7 Nump

Quantifier phrases are coded as nump, e.g., telephone numbers and other quantifications in speech.

3.8 Pp

Pp stands for prepositional phrase, and this category comprises prepositional phrases. Even if a prepositional phrase constitutes an adverbial it should be coded as a prepositional phrase.

3.9 Vp

Complete verb phrases are coded as vp, i.e., verbs which are combined with other words, or several verbs. Individual intransitive verbs should be coded as v under the parts_of_speech schema.

3.10 Subordinate clause

Complete subordinate clauses are coded as subordinate clause.

When we encounter utterances or parts of utterances which begin with a coordinating or subordinating conjunction, we must check to see whether it is a case of coordination with an earlier utterance or of subordination. In the case of coordination, the conjunction should be coded separately under parts_of_speech–conj and the sentences separately as complete sentences. The same goes for coordination of Vp or Np. In the case of subordination, subordinate clauses should be combined with their subordinating conjunctions into a larger unit and coded as “subordinate_clause” under “phrases”.

4. The coding schema *parts_of_speech.pl*

4.1 Coding schema

The coding schema parts_of_speech.pl contains the following categories:

adj/1—adjective
adv/1—adverb
art/1—articles
conj/1—conjunctions (coordinating and subordinating)
fb/1—feedback-words, see 3.5
inf/1—infinitive markers
interj/1—interjections
n/1—nouns
num/1—quantifiers
ocm/1—OCM-words
part/1—particles
prep/1—prepositions
pron/1—pronouns
v/1—verbs

5. Coding OCM in MaxGram

5.1 What is coded as OCM in MaxGram?

In the OCM manual there are examples of many different OCM phenomena which occur in different constructions, e.g., repetitions, hesitation sounds, and pauses. When we code MaxGram, only particular words which always (or often) have an OCM function should be coded as OCM. These include “e1” *(uh)*, “m1” *(um)* and other hesitation sounds. “Liksom” *(like)* often has an OCM function. These words are coded under the parts_of_speech -schema and are marked as OCM.

NOTE that feedback words often have an OCM function, but in MaxGram they should always be coded as feedback (fb), regardless of function. Nor should pauses be coded as OCM, but rather as parts of larger structures.

5.2 Coding OCM within sentences

When something which otherwise has the structure of a complete sentence contains OCM expressions, we code the sentence as a sentence and subsequently mark OCM inside the sentence. We do not divide a sentence into smaller parts because someone stumbles over his words or repeats himself in the middle of the sentence.

An example of this:

<1 ja1 tänker mej nån som skrive <2 e1>2 dikter om0 naturen>1
<1 I am thinking of someone who writes <2 uh>2 poems about nature>1

where <1> is a declarative sentence and <2> OCM.

5.3 Coding OCM within phrases

The same rule that applies for coding OCM within sentences applies to OCM phenomena within phrases. That is, we code OCM within phrases as well. This can take the following form:

$R: <1 en riktig <2e1>2 slutsats>1
$R: <1 a real <2uh>2 conclusion>1

where <1> is an np and <2> is an OCM word.

5.4 Coding OCM initially in phrases

If an OCM phenomenon occurs initially in a phrase, we code it as belonging to the phrase. For example:

A: <1<2 e1>2 lite grann åt de0 där hållet igen>1
A: <1<2 uh>2 a little bit in that direction again>1

where <1> is an advp and <2> OCM.

6. Coding feedback in MaxGram

6.1 Fb-words which do not have a feedback function

When we code MaxGram, all feedback words should be coded as feedback, regardless of whether they appear to have another function, e.g., OCM. However, they should not be coded above sentences, as one does with OCM. If a feedback word is part of a sentence, then only the sentence as a whole should be coded.

6.2 Utterances consisting of feedback plus a sentence

In utterances like “ja0 de0 vill ja1” *(yes, I do want that)*, “ja0” should be coded on its own as fb and “de0 vill ja1” separately as a declarative sentence. This applies even if the feedback is at the end of the utterance. Note that this feedback can also be feedback-elicitors like “eller” *(right)*, “väl” *(surely)* och “eller hur” *(isn’t that so)*.

7 Other principles for MaxGram coding

7.1 Direct speech

Direct speech can be seen as part of a sentence, as in the example below.

$A: ja1 kan tänka mej att en0 urneurotisk person säjer / å4 ja1 har ju en0 naturli
$A: I can imagine that a highly neurotic person would say / oh of course I have a natural

känsla av att de0 e0 si å0 så0 va0
feeling that it's not so good, eh

is thus a declarative sentence.

7.2 Utterances consisting of a sentence plus a name or noun

In the case of utterances consisting of a name plus a sentence, one should discern whether the name is used in direct address or not. If used in direct address, it should not be coded as part of the sentence. For example, in the utterance

"va1 tror du om den analysen Jonas"
(what do you think of that analysis, Jonas)

"va1 tror du om den analysen" is coded as a wh-question and "Jonas" as a noun, external to the question. However, if someone says, "Astrid Lindgren she's a fine author," then everything should be coded as a declarative sentence. This also applies in sentences with pronouns, e.g., "de0 va1 de0 de0" *(That it was.)*.

In other cases where a noun precedes a self-contained, complete sentence, the noun is to be seen as part of the following sentence. An example:

$G: respekt de0 hänger no också mer ihop me0 auktoriteter
$G: respect that probably goes along more with authority

Here the entire utterance is coded as a declarative sentence. Hence one should not code the noun separately and the rest as a sentence.

7.3 When a word is omitted

Sometimes a word is not transcribed, even though it has been said—it has "fused together" with another word. An example here is "de0 kallt ida" *('s cold today)*. "de0" stands for "det är" *(it is)* and should also be coded as such. The sentence above thus becomes a declarative sentence. This does not apply in just any case where a word has been skipped, but only when it has "fused together" with some other word, i.e., it does not sound as if any word is missing.

7.4 Foreign words

All foreign words should be coded with the word-class they would receive if they were Swedish words.

7.5 Interrupted words

If it is possible to guess which word has been interrupted, it should be coded as if it were that word. If it is not possible to guess, then it should be coded as OCM.

7.6 When the transcriber has made an error

If one discovers that the transcriber obviously has transcribed incorrectly, one should make a note about it and code it as it stands for the time being, i.e., **not** as one believes it should be! In order to correct these errors, a special procedure must be followed, and one must therefore not attempt it oneself, but rather pass on all error notes to someone who has learned how to make corrections. Ask whoever is responsible for the coding.

7.7 Words within parentheses

Words in parentheses, i.e., which the transcriber was unsure about, should be coded as if the parentheses were not there.

References

Allwood, J. (1976) Linguistic Communication as Action and Cooperation. *Gothenburg Monographs in Linguistics 2*. Göteborg University, Department of Linguistics.

Allwood, J. (1977) A Critical Look at Speech Act Theory. In Dahl (Ed.). *Logic, Pragmatics and Grammar*, Lund, Studentlitteratur, pp. 53-69.

Allwood, J. (1978) On the Analysis of Communicative Action. In Brenner (Ed.) *The Structure of Action*. Oxford: Basil Blackwell.

Allwood, J. (1987) A Semantic Analysis of Understanding. In V. Rosén (Ed.) *Papers from the Tenth Scandinvian Conference of Lingustics*, University of Bergen, Department of Linguistics and Phonetics.

Allwood, J. (ed) (1988a) Feedback in Adult Language Acquisition (Final Report II). *Ecology of Adult Language Acquisition* (ESF).

Allwood, J. (1988b) "Om det svenska systemet för språklig återkoppling". In Linell, P., Adelswärd, V., Nilsson, T. and Pettersson, P. A. (eds) *Svenskans beskrivning* 16. SIC 21a. University of Linköping: Department of Communication Studies.

Allwood, J. (1994) "Obligations and Options in Dialogue". In *Think* , vol 3, May 1994, ITK, Tilburg University.

Allwood, J. (1995) An Activity-based Approach to Pragmatics. *Gothenburg Papers in Theoretical Linguistics, 76*. Göteborg University, Department of Linguistics. Forthcoming in Bunt and Black (Eds.) *Approaches to Pragmatics*.

Allwood, J., Nivre, J. and Ahlsén, E. (1990) "Speech Management: On the Non-Written Life of Speech". *Gothenburg Papers in Theoretical Linguistics, 58*. University of Göteborg, Department of Linguistics. Also in *Nordic Journal of Linguistics*, 13.

Allwood, J., Nivre, J. and Ahlsén, E. (1992) On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics* 9, 1–26.

Austin, J. A. (1962) *How to Do Things with Words*. Harvard University Press.

Bühler, K. (1934) *Sprachtheorie*. Jena: Fischer.

Grice, H. P. (1975) Logic and conversation. In P. Cole and J. L. Morgan (Eds.) *Syntax and Semantics Vol. 3: Speech Acts*. New York: Seminar Press, 41–58.

Kowtko, J. and Isard, S. (1991) Conversational games within dialog. *Proceedings of the ESPRIT Workshop on discourse Coherence*, University of Edinburgh, 4-6 April, 1991.

Larsson, S. (1997): *TRACTOR Användarmanual*. Department of Linguistics, Göteborg University.

Moeschler, J. (1989) *Modélisation du dialog*. Paris: Hermès.

Nivre, J. (1999a): *Modifierad Standardortografi Version 6*. Department of Linguistics, Göteborg University.

Nivre, J. (1999b): *Transcription Standard Version 6.2*. Department of Linguistics, Göteborg University.

Sadek, M. D. (1991) Dialogue acts are rational plans. In *Proceedings of the ESCA/ETR workshop on multi-modal dialogue*.

Schegloff, E. and Sacks, H. (1973) Opening up closings. *Semiotica*, 7.4, 28-327.

Sinclair, J. and Coulthard, R.M. (1975) *Towards and Analysis of Disourse. The English Used by Teachers and Pupils*. London: Oxford University Press.

Recommended Reading

Abelin, Åsa & Allwood, Jens (1998) *Jämförelse mellan OCM kodningsstandard och Robert Eklunds disflueringskodningsstandard* (Internal report). Göteborg: Göteborg University, Department of Linguistics.

Allwood, Jens (Ed.) (1996 and later editions) Talspråksfrekvenser, Ny och utvidgad upplaga. *Gothenburg Papers in Theoretical Linguistics S21*. Göteborg: Göteborg University, Department of Linguistics.

Allwood, Jens (1999) Some frequency based differences between spoken and written swedish. In *Proceedings of the 16th Scandinavian Conference of Linguistics*. Turku: University of Turku, Department of Linguistics.

Allwood, Jens (2000) *Expressive and Evocative Functions and Obligations. Coding Manual*. Version 1.0. Göteborg University, Department of Linguistics.

Allwood, Jens & Björnberg, Maria (1999) *Coding Schemas within the SDS Project—A Comparison*. http://www.ling.gu.se/SLSA/Documents/coding_schema_comparison.ps

Allwood, Jens & Hagman, Johan (1994) Enkla mått på samtal. In F. Gregersen & J. Allwood (Eds.), *Spoken Language, Proceedings of the XIV Conference of Scandinavian Linguistics*.

Allwood, Jens, Abelin, Åsa & Grönqvist, Leif (1998) *Kort beskrivning och jämförelse av transkriptionssystem från Lund, Telia, Linköping, och Göteborg*. http://www.ling.gu.se/~leifg/doc/jfrelse_transkriptionssystem.pdf

Clark, H. H. and Schaefer, E. F. (1989) Contributing to discourse. *Cognitive Science,* 13:259–294.

Dahlbäck, Nils & Jönsson, Arne (1998) *A coding manual for the Linköping dialog model.* Linköping: Linköping University, Department of Computer and Information Science.
Nivre, Joakim (1999) *Transcription Standard. Version 6*. Göteborg: Göteborg University, Department of Linguistics.

Hagman, J. and Nivre, J. (1993) *Transcription Standards*. University of Göteborg: Department of Linguistics.

Lager, T. (1992) *TagLog—the manual.* University of Göteborg: Department of Linguistics.

Nivre, Joakim & Grönqvist, Leif (1999) Tagging a corpus of spoken Swedish. Forthcoming in *International Journal of Corpus Linguistics*. http://www.ling.gu.se/SLSA/Documents/sot.ps

Nivre, Joakim, Tullgren, Kristina, Allwood, Jens, Ahlsén, Elisabeth, Holm, Jenny, Grönqvist, Leif, Lopez-Kästen, Dario & Sofkova, Sylvana (1998) Towards multimodal spoken language corpora: TransTool and SyncTool. *Proceedings of ACL-COLING 1998*, June 1998. http://www.ling.gu.se/~leifg/doc/COLING98.pdf

Research on Language and Social Interaction. Karen Tracy, ed. Hillsdale, N.J.: Lawrence Erlbaum Associates, Inc.

Van Hout, Roeland & Rietveld, Toni (1993) *Statistical Techniques for the Study of Language and Language Behaviour*. Berlin & New York: Mouton de Gruyter.