# Comparing methods in natural language generation with cognitive processes in language production

Course held at the
Summer School in Computational Linguistics
University of Zadar 2010

Ralf Klabunde

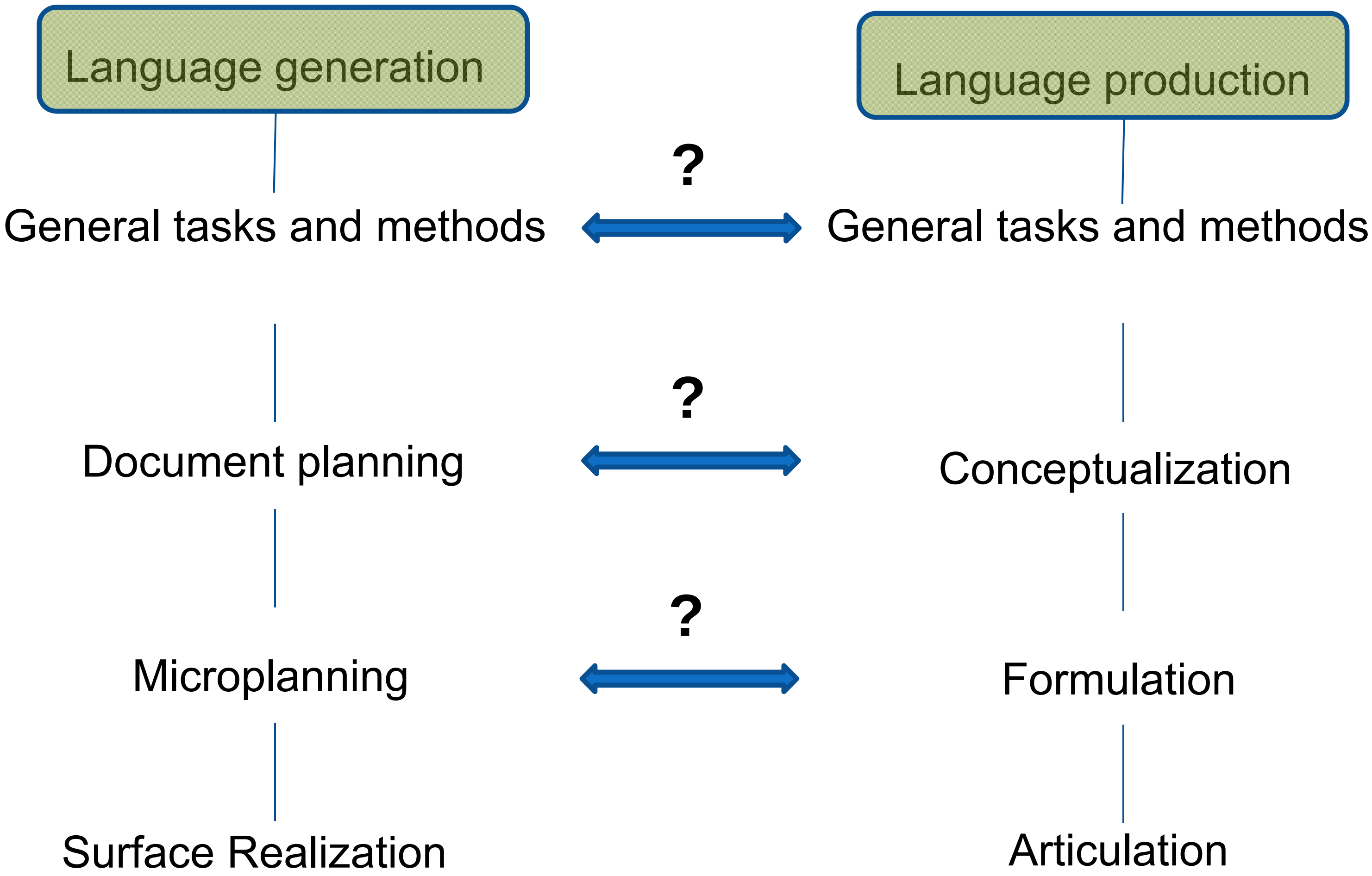
# Issues

- How does a computer transform data/knowledge into a text?
- What are the standard methods in natural language generation?

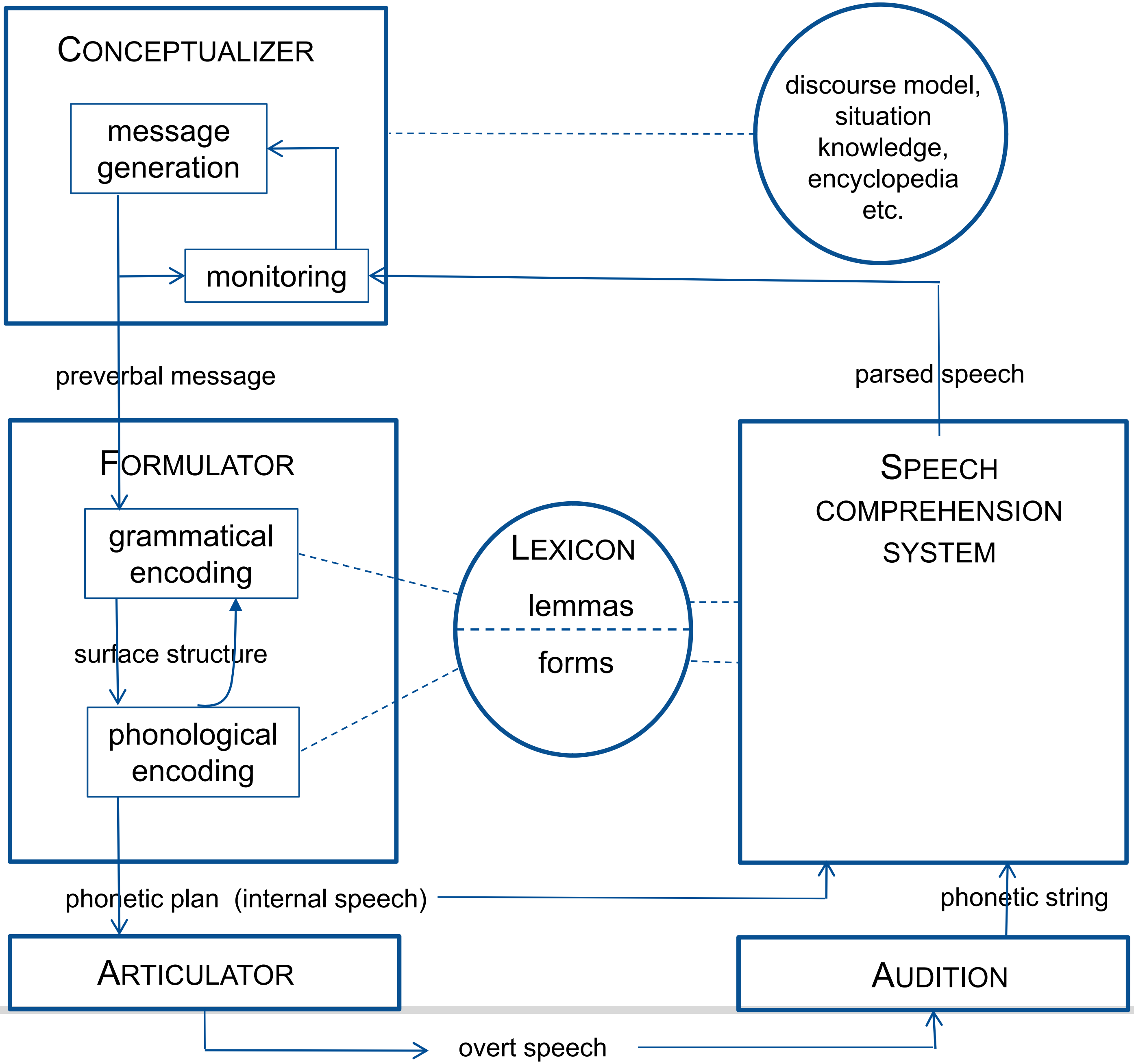
- How do speakers transform their intention into an utterance?
- What are the basic processes in human language production?
- What do we know about the interaction between modules in language production?

- How might research on natural language generation (NLG) benefit from insights on human language production?

# Outline of this course

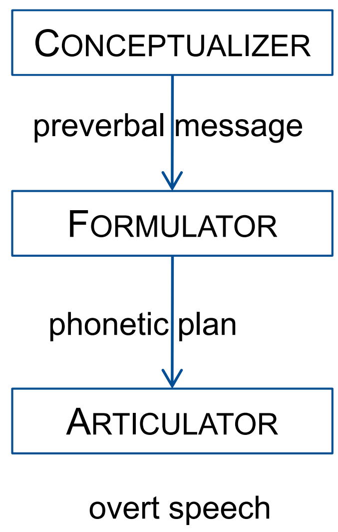


# Language production: A blueprint of the speaker (Levelt, 1989)



# Properties of Levelt's model

- Modularity: components of the model as autonomous specialists
- Characteristic output of each module
- Unidirectional information flow
- Incremental (i.e. semi-parallel) processing



# An example: talking about the weather

- A: "It doesn't look like it's going to stop raining today."
- B: "Well, at least my lawn will be happy"

Apparently, A wants to start a conversation.
I want to signal that I am ready for a conversation.

CONCEPTUALIZER

preverbal message

`resume_narrative(minimal_positive_event(lawn(x) ∧ of(x,speaker) ∧ happy(x)))`

FORMULATOR

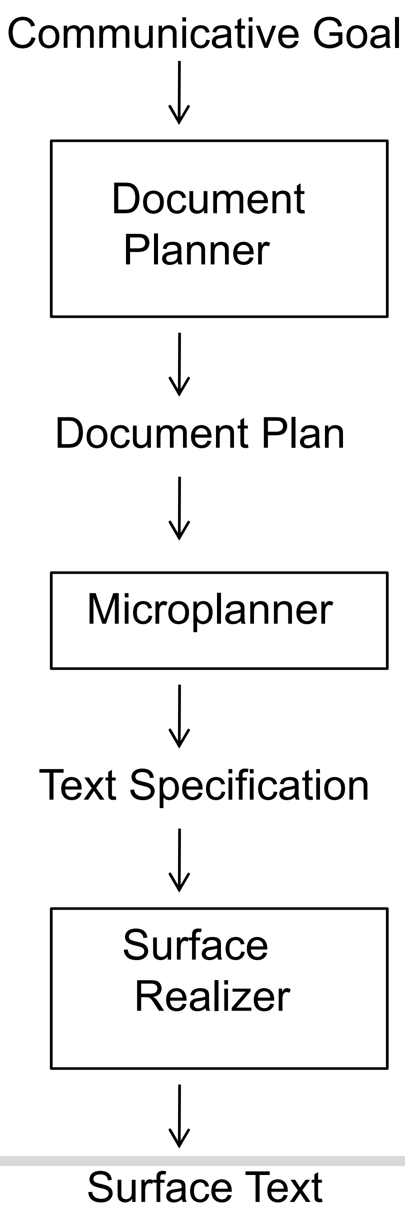
| well | at least | my lawn | will be happy |
| --- | --- | --- | --- |
| [wɛl | æt liːst | maɪ lɔːn | wɪl biː ˈhæpi] |

phonetic plan

ARTICULATOR

overt speech

# A reference architecture for generation systems



## Natural language generation is *choice*:

| Task | Choice |
|---|---|
| Content determination | Choosing the appropriate content to express from a potentially over-specified input |
| Document structuring | Deciding how chunks of content should be grouped in a document and how chunks should be related in rhetorical terms |
| Lexical selection | choosing the lexical items that are most apppropriate for expressing particular concepts |
| Generation of referring expressions | choosing the right linguistic form for reference to the objects to be discussed: pronouns, (in)definite NPs |
| Sentence aggregation | choosing the appropriate chunks for the selected content: phrases, clauses, or sentences |
| Linguistic realization | Converting abstract representations of sentences into real text |
| Structure realization | Converting abstract structures (e.g., paragraphs, sections) into mark-up symbols for the document presentation component |

# Natural language generation is *choice*:

| Module | Content task | Structure task |
|---|---|---|
| Document planning | Content determination | Document structuring |
| Microplanning | Lexicalisation<br>Referring expression generation | Aggregration |
| Realization | Linguistic realization | Structure realization |

## An example: Generating a weather report

- Content selection from raw input data:

```
96, 122, 1, 5, 6.34, 723, -11, -2.003, -2.032, .445, 1024, 17.72,
11.21, 20.09, -15.72
96, 122, 1, 5, 6.25, 615, -14, -2.894, -1.293, .669, 1022, 14.92,
15.36, 19.88, -12.48
....
```

- Several thousand data records must be merged and simplified in a daily weather record (next slide)

- Resulting text:

  Summary: The month was rather dry with only three days of rain in the middle of the month. The total for the year so far is very depleted again, after almost catching up during march.

# An example daily weather record

- type: dailyweatherrecord
- date:
  - day: 1
  - month: 5
  - year: 1996
- temperature:
  - minimum:
    - unit: degreesCentrigrade
    - number: 15
  - maximum:
    - unit: degreesCentigrade
    - number: 26
- rainfall:
  - unit: millimetres
  - number: 4

# Outline of this course

| Language generation | ? | Language production |
|---|---|---|
| General tasks and methods | ↔ | General tasks and methods ✓ |
| Document planning | ? ↔ | Conceptualization |
| Microplanning | ? ↔ | Formulation |
| Surface Realization | | Articulation |

## How do we proceed?

- We take the different stages in NLG and
  - We will have a brief look on the basic mechanisms used
  - Then we will have a brief look on some new approaches
  - Finally, we compare these approaches with psycholinguistic insights on human language production

# Document planning: Content determination and document structuring

- Basic mechanisms:
  - Content determination:
    - Defining the content determination rules requires a corpus in order to determine user requirements
    - Assembling an initial corpus of output texts
    - Analysing the information content of these texts
      - Classifying each constituent/sentence into one of the following categories:
        - Is a textual constituent always present in the output corpus?
        - Is information in the texts directly available in the data?
        - Is the information presented in the texts computable from the input data?
        - Is the information in the texts not available?
    - Finally, a target text corpus is created

# Document planning: Content determination and document structuring

- Basic mechanisms:

  - Example:
    - Taking existing weather reports
    - Analysing these texts:
      - Unchanging text: "summary:" is always present
      - Directly available data: Reporting the amount of rainfall on the wettest day in a given month
      - Computable data: *The total for the year so far is very depleted again* (straightforward computation), *after almost catching up during March* (requires looking for patterns in the data across several month)
      - Unavailable data: for example, mentioning zodiac signs
    - Creating the texts-to-be-generated

# Document planning:  Content determination and document structuring

- Basic mechanisms:

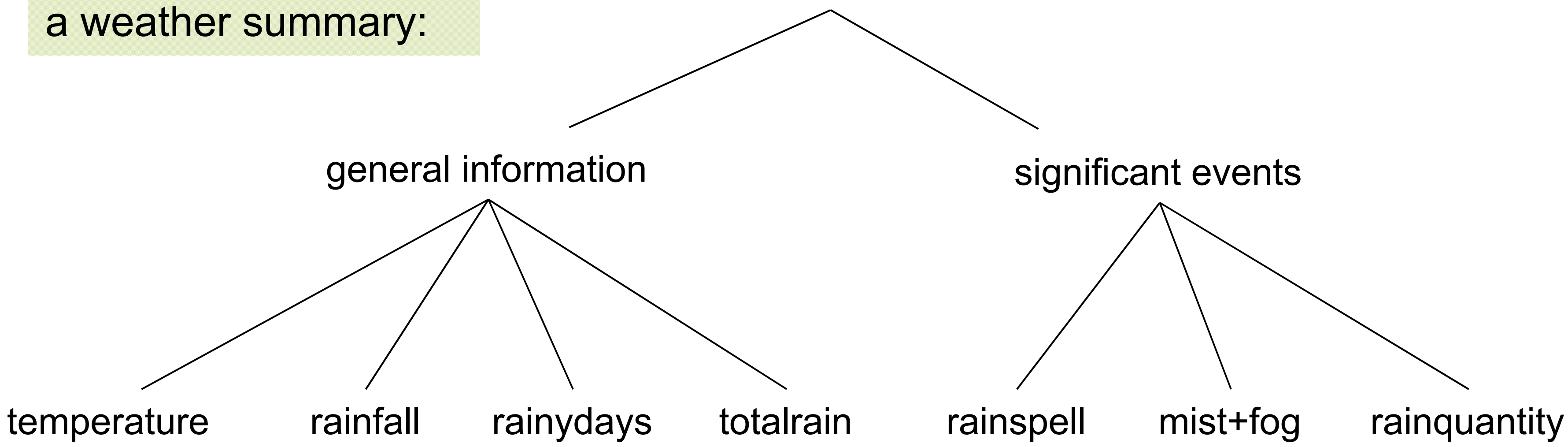
  - Document structuring: An (ideal) document planner takes as input a quadruple ⟨k,c,u,d⟩ with:
    - k: the knowledge source to be used
    - c: the communicative goal to be achieved
    - u: a user model
    - d: a discourse history

  - To construct a document plan, the planner orchestrates three activities:
    - Constructing messages from k
    - Deciding which messages need to be communicated in order to satisfy c
    - Carrying out document structuring to organize the presentation of these messages in a coherent text

  - Document structuring is the most application-dependent aspect of NLG

# Document planning:  Content determination and document structuring

A simple structure of
a weather summary:




The month was cooler and drier than average, with the average number of rain days. The total rain for the year so far is well below average. There was rain on every day for eight days from the 11th to the 18th, with mist and fog patches on the 16th and 17th. Rainfall amounts were mostly small, with light winds.

# Document planning: Content determination and document structuring

- The notion of a MESSAGE forms a useful abstraction that mediates between the data structures and the texts to be generated

- Key idea: for any given domain of application, we define a set of entities, concepts and relations that characterizes the domain in terms that can easily be mapped into linguistic forms

# Document planning: Content determination and document structuring

## Two example messages

- type: MonthlyTemperatureMsg
- period:
  - month: 05
  - year: 1996
- temperature:
  - type: RelativeVariation
  - magnitude:
    - unit: degreesCentigrade
    - number:2
  - direction: --

- type: MonthlyRainfallMsg
- period:
  - month: 05
  - year: 1996
- rainfall:
  - type: RelativeVariation
  - magnitude:
    - unit: millimetres
    - number: 2
  - direction: --

# Document planning: Content determination and document structuring

- type: dailyweatherrecord
- date:
  - day: 17
  - month: 05
  - year: 1995
- temperature:
  - minimum:
    - unit: degreesCentigrade
    - number: 10.1
  - maximum:
    - unit: degreesCentigrade
    - number: 21.3
- rainfall:
  - unit: millimetres
  - number: 12

A daily weather record

- type: DPDocument
- title
  - type: PSCannedText
  - text: Weather summary for July 1996
- children:
  - type: constituent set
  - relation: sequence
  - constituent1:
    - type: DocumentPlan
    - children: (See DP below)
  - constituent2:
    - type: DocumentPlan
    - children:
      - type: ConstituentSet
      - relation: narrativeSequence
      - constituent1: (RainEventMsg for 27th)
      - constituent2: (RainEventMsg for 28th)

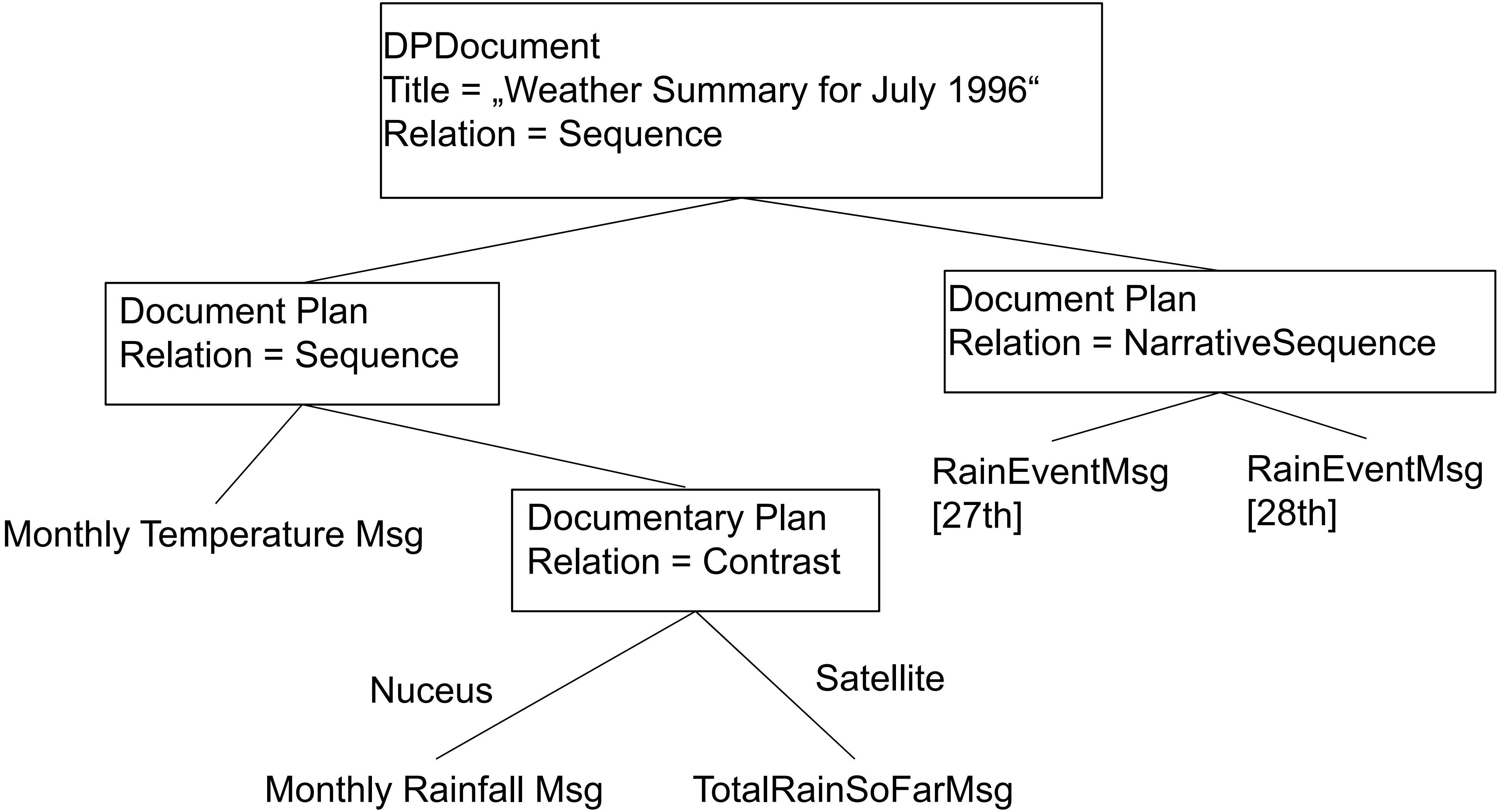
Document plan for this text:

**The month was slightly warmer than average with almost exactly the average rainfall, but rainfall for the year is still very depleted. Heavy rain fall on the 27th and 28th.**

DP =

- type: ConstituentSet
- relation: sequence
- constituent1: (MonthlyTemperatureMsg)
- constituent2:
  - type: DocumentPlan
  - children:
    - type: NSConstituents
    - nucleus: (MonthlyRainfallMsg)
    - satellite1:
      - type: SatelliteSpec
      - relation: contrast
      - satellite: (TotalRainSoFarMsg)

## Schematic version of the document plan



# Document planning: Content determination and document structuring

- What constitutes a message?
  - A message is a kind of semantic representation that expresses configurations of domain elements:
    - Facts from the input database or the knowledge base
    - Information expressible in simple phrases up to multisentential text
  - Message granularity depends on the types of variations expected in the output texts
  - A weather report example:
    - if temperate and total rainfall are always mentioned in the same clause: one message
    - If temperate and total rainfall are mentioned different text parts: two messages

# Document planning: Content determination and document structuring

- What constitutes a message?
  - Weather reports require the following messages:
    - Messages for standard reports:
      MONTHLYRAINFALLMSG
      MONTHLYTEMPERATUREMSG
      TOTALRAINSOFARMSG
      MONTHLYRAINYDAYSMSG
    - Messages for significant events:
      RAINSPELLMSG
      TEMPERATURESPELLMSG
      RAINEVENTMSG
      TEMPERATUREEVENTMSG

# Document planning: Content determination and document structuring

- What constitutes a message?
  - Let's have a look at the message type MONTHLYRAINFALLMSG:
    - It is the base for the generation of clauses like *Rainfall was well above average*
    - Talking about the parameter "rainfall" can be done in several ways:
      - Comparing it to the average value for this month over the period of record
      - Result: phrases like *well above average, drier than average* etc.
      - Identifying it as having a significant value (*the rainiest instance of this month on record, the rainiest month overall*, etc.)

# Example MonthlyTemperatureMsg and RainEventMsg messages

type: MonthlyTemperatureMsg
period: [month: 07, year: 1996]
temperature: [
  type: RelativeVariation
  magnitude: unit: degreesCentigrade
             number: 2
  direction: +
]

type: RainEventMsg
period: [month: 07, year: 1996]
date: [day: 27, month: 07, year: 1996]
rainType: heavy

# Document planning: Content determination and document structuring

- What constitutes a message?
  - The message type TEMPERATURESPELLMSG:
    - Such a message will be constructed whenever the daily temperature remains within one of these bands for some sequence of days:

| temperature | symbolic value |
|---|---|
| over 40.0 | extremely hot |
| 35.0 to 39.9 | veryhot |
| 30.0 to 34.9 | hot |
| 25.0 to 29.9 | verywarm |
| 20.0 to 24.9 | warm |
| 15.0 to 19.9 | mild |
| 10.0 to 14.9 | cool |
| 05.0 to 09.9 | cold |
| 00.0 to 04.9 | verycold |
| below 00.0 | freezing |

# A TemperatureSpellMsg message

- type: TemperatureSpellMsg
- period:
  - year: 1996
  - month: 06
- spell:
  - begin:
    - day: 06
    - month: 06
    - year: 1996
  - end:
    - day: 09
    - month: 06
    - year: 1996
- temperature: extremelyhot

# Document structuring

- The key concepts in any discussion of document structuring are rhetorical relations (discourse relations)

- These key concepts are borrowed from Rhetorical Structure Theory (RST; Mann & Thompson,1988)
  - A text is coherent by virtue of the relationships that hold between the constituent elements of that text

- In NLG, rhetorical relations specify the relationships that hold between messages or groups of messages

- Most of these relations are binary; their message groups have two components

# Document structuring

- Examples:

- *I like to collect old Fender guitars.*
  *My favourite instrument is a 1951 Stratocaster.*

  Elaboration

- *I like to collect old Fender guitars.*
  *However, my favourite instrument is a 1991 Telecaster.*

  Contrast

  cue word to signal contrast

# Document structuring: RST

- RST claims 25 rhetorical relations
- These relations hold between so-called text spans
- Most RST relations are binary, they hold between a nucleus and a satellite
- A relation definition has four parts:
  - Constraints on the nucleus
  - Constraints on the satellite
  - Constraints on the combination of nucleus and satellite
  - The effect (on the reader)

| Relation name | Constraints on either S or N | Constraints on N + S | effect |
|---|---|---|---|
| background | on N: R won't comprehend N sufficiently before reading text of S | S increases the ability of R to comprehend an element in N | R's ability to comprehend N increases |
| elaboration | none | S presents additional detail about the situation or some element of subject matter which is presented in N or inferentially accessible in N in one or more of the ways listed below. In the list, if N presents the first member of any pair, then S includes the second:<br>set :: member<br>abstraction :: instance<br>whole :: part<br>process :: step<br>object :: attribute<br>generalization :: specific | R recognizes S as providing additional detail for N. R identifies the element of subject matter for which detail is provided. |

# The RST definition of Elaboration

*relation name:* Elaboration
*constraints on N*: none
*constraints on S*: none
*constraints on N+S combination*:

S (the satellite) presents additional detail about the situation or some element of the subject matter which is presented in N (the nucleus) or inferentially accessible from N in one or more of the following ways:

1. N: set; S: member of set
2. N: abstraction; S: instance
3. N: whole; S: part
4. N: process; S: step
5. N: object; S: attribute of object
6. N: generalisation; S: specific instance

*the effect*: R (the reader) recognises the situation presented in S as providing additional detail for N. R identifies the element of subject matter for which detail is provided.

*the locus of the effect*: N and S

# Discourse Planning

- RST-definitions can be decoded as planning operators:
  - Constraints are modelled as preconditions of operators
  - Effects are mapped into postconditions

- The approach of Hovy (1990): a top-down refinement planner

- Application domain: multimodal database information display system: answering user‘s request for the display of information from a data base of naval information

## Discourse Planning

- Communicative goal:
  `(GOAL (BMB SPEAKER HEARER (SEQUENCE-OF E1 ?NEXT)))`
- Selected input:

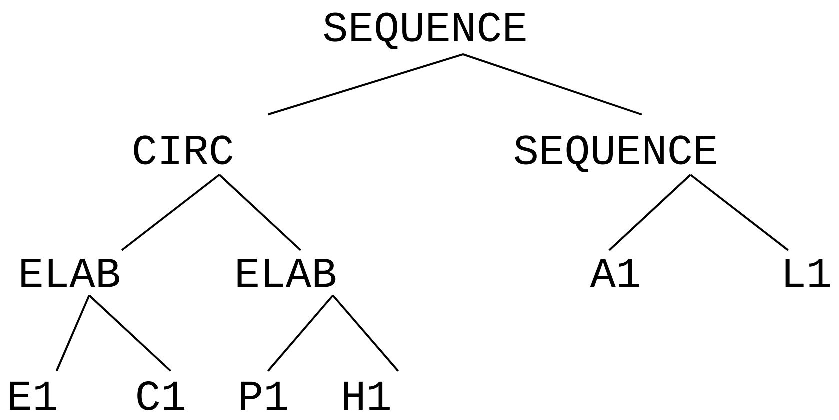
```
((ENROUTE E1)
 (ACTOR E1 K1)
(DESTINATION E1 S1)
(NEXT-ACTION E1 A1)
(LOCATION E1 P1))
((ARRIVE A1)
(ACTOR A1 K1)
(TIME A1 T1)
(NEXT-ACTION A1 L1))
((READINESS-STATUS C1)
(NAME C1 C4))

((POSITION P1)
(HEADING P1 H1)
(LATITUDE P1 79)
(LONGITUDE P1 18))
((HEADING H1)
(COURSE H1 195))
((LOAD L1)
(ACTOR L1 K1)
(STARTTIME L1 T2)
(ENDTIME L1 T3))

((SHIP K1)
(NAME K1 KNOX)
(READINESS K1 C1))
((PORT S1)
(NAME S1 SASEBO))
((DATE T1)
(DAY T1 24)
(MONTH T1 4))
((DATE T2)
(DAY T2 25)
(MONTH T2 4))
((DATE T3)
(DAY T3 28)
(MONTH T3 4))
```

# Discourse Planning

- Resulting paragraph structure (left branches: nuclei, right branches: satellites)




- Corresponding text:
  Knox, which is C4, is en route to Sasebo. It is at 79N 18E heading SSW. It will arrive on 4/24, and will load for four days.

# Discourse planning

- Basic idea:
  - RST relations as plan operators
  - Nucleus and satellite requirements as semantic preconditions
  - Introduction of growth points of subgoals that are permitted by coherence
  - Effects are represented by the beliefs of the interlocutors
    BMB (believe mutual believe): `(BMB X Y P)` means: P follows from X's beliefs about what X and Y mutually believe

# The sequence relation

```
Name: SEQUENCE
Results:
((BMB SPEAKER HEARER (SEQUENCE-OF ?PART ?NEXT)))
Nucleus requirements/subgoals:
((BMB SPEAKER HEARER (TOPIC ?PART)))
Satellite requirements/subgoals:
((BMB SPEAKER HEARER (TOPIC ?NEXT)))
Nucleus+Satellite requirements/subgoals:
((NEXT-ACTION ?PART ?NEXT))
Nucleus growth points:
(((BMB SPEAKER HEARER (CIRCUMSTANCE-OF ?PART ?CIR))
 (BMB SPEAKER HEARER  (ATTRIBUTE-OF ?PART ?VAL))
 (BMB SPEAKER HEARER  (PURPOSE-OF ?PART ?PURP)))
Satellite growth points:
(((BMB SPEAKER HEARER (ATTRIBUTE –OF ?NEXT`?VAL))
 (BMB SPEAKER HEARER (DETAILS-OF ?NEXT ?DETS))
(BMB SPEAKER HEARER  (SEQUENCE-OF ?NEXT ?FOLL)))

Order: (NUCEUS SATELLITE)
Relation-phrases: („“ „then“ „next“)
```

speaker and hearer will mutually believe that the relation SEQUENCE-OF holds between two input entities

The nucleus is bound to the variable ?PART, the satellite to the variable ?NEXT

?PART precedes ?NEXT

Suggestions for additional material for the nucleus: circumstantially related material (time, location, etc.), attributes (size, color, etc.), purpose;

analogously the satellite

The order is: nucleus first; using either no cue word, or „then“, or „next“

# The discourse planner

- The planner searches for input entities that match the requirements of the nucleus and the satellite
- If fulfilled, the planner considers the growth points: recursive search for relations and matching their nucleus and satellite requirements to the input.
- The planning process bottoms out when either
  - all input entities have been incorporated into the tree, or
  - no extant goals can be satisfied by the remaining input entities

## The discourse planner

- The resulting text structure provides constraints on the realization of the text:
  - Use of cue words
  - Satellite of the ELABORATION relation is realized as relative clause
  - Information in the satellite of a temporal SEQUENCE relation is used in the future tense

"Knox, which is C4, is en route to Sasebo. It is at 79N 18E heading SSW. It will arrive on 4/24, and will load for four days."

# An alternative: The use of schemas

- Sometimes texts follow regular patterns
- An example:
  - A weather report starts with the month‘s overall temperature and rainfall
  - then the month will be compared with the same month in previous years
  - then significant events will be described (periods of extreme temperature or rainfalls)
- For these kinds of text types, schema-based approaches to discourse structuring are useful as well

# An alternative: The use of schemas

- Schemas are patterns that specify how a document should be constructed from messages or instantiations of other schemas
- Schemas often include optional elements with associated conditional tests
- A set of schemas constitutes a kind of text grammar
  - Example: PEBA (Milosavljevic & Dale, 1996)

# An alternative: The use of schemas

- PEBA compares two entities
- PEBA's compare-and-contrast schema:

```
CompareAndContrast →
        DescribeLinnaeanRelationship  CompareProperties
CompareProperties →
        CompareProperty  CompareProperties
CompareProperties → ∅
```

- Each text defines the two entities in terms of the Linnaean taxonomy, and then compares the properties of both entities
- The terminal symbols in the grammar call computational procedures that examine the data source to locate information that can be included in the text

# An alternative: The use of schemas

- Two simple schemas for weather reports:

```
Schema DescribeWeather() returns DocumentPlan;
  DocumentPlan DP1 = DescribeMonthOverall();
  DocumentPlan DP2 = DescribeSignificantEvents();
  return NewDocumentPlan(DP1, DP2, Sequence);
  end Schema.

Schema DescribeMonthOverall() returns DocumentPlan;
  Message M1 = getMessage(MonthlyRainfallMsg);
  DocumentPlan DP1 = DescribeOverallPrecipitation();
  return NewDocumentPlan(M1, DP1, Sequence);
  end Schema.
```

# An alternative: The use of schemas

- Schemas as text grammars expand an initial schema in a top-down manner
- A more flexible approach is a bottom-up procedure: messages are combined until a discourse plan has ben achieved
- All bottom-up approaches proposed in the literature assume:
  - All messages selected by the content determination procedure must be included in the discoure plan
  - There is a means to determine what discourse relation holds between two messages

# An alternative: The use of schemas

- The bottom-up discourse structuring algorithm of Reiter & Dale (2000:108)

Let POOL = *messages produced by content determination mechanism*

**while** (size(POOL) ≥ 1) **do**

find all pairs of elements in POOL which can be linked by a discourse relation

assign each such pair a desirability score, using a heuristic preference function

find the pair $E_i$ and $E_j$ with the highest prefence score

combine $E_i$ and $E_j$ into a new DocumentPlan $E_k$, using an appropriate discourse relation;

remove $E_1$ and $E_j$ from POOL and replace them with $E_k$;

**end while**

## Bottum-up construction of a document plan

Example:

**Step 1** POOL is initialised to contain $E_1$, $E_2$ and $E_3$.

**Step 2** Two elements are selected from POOL and combined. There are two possibilities: We could combine $E_2$ and $E_3$ under the Sequence Rule, with a score of 1; or we could combine $E_2$ and $E_3$ under the Contrast Rule, with a score of 2. Since the latter has the higher score, these elements are combined. The result is a new DocumentPlan $E_4$, whose nucleus is $E_2$, which replaces the constituent elements in the pool.

**Step 3** POOL now contains $E_1$ and $E_4$. Since $E_2$ is the nucleus of $E_4$, these can be combined via the Sequence Rule, producing a new document plan $E_5$.

**Step 4** POOL now contains just one element, E5, so the document structuring process terminates.

• **Elaboration Rule**. A TotalRainfallMsg can be added as an Elaboration to a MonthlyRainfallMsg if both messages have the same value for Direction (the direction of variation).

• **Contrast Rule.** A TotalRainfallMsg can be added as a Contrast to a MonthlyRainfallMsg if the two messages have different Direction values.

• **Sequence Rule.** A MonthlyTemperatureMsg (or a DocumentPlan whose nucleus is a MonthlyTemperatureMsg) and a MonthlyRainfallMsg (or a DocumentPlan whose nucleus is a MonthlyRainfallMsg) can be combined using the Sequence relation.

# Interaction between content determination and discourse planning

- Two alternatives:
  - Data-driven process: Content determination identifies the messages, discourse structuring process combines them into a coherent text
  - Hypothesis-driven process: The discourse structuring process starts with a notion how a text would look like and requests a content determination procedure to find/create appropriate messages

## Interaction between content determination and discourse planning

| Content determined … | | |
|---|---|---|
| | before document structuring (*data-driven*) | during document structuring (*hypothesis-driven*) |
| Relations-based | Text Structurer; Hovy (1988) | PEA; Moore (1994) |
| Schema-based | TEXT, McKeown (1985) | ModelExplainer, Lavoie et al. (1997) |

## Summary

- Two approaches to discourse structuring:
  - Plan-based:
    - RST as descriptive model
    - RST-based relations as planning operators
    - Use of growth points for plan expansion
  - Schema-based:
    - Use of schemata for text patterns
    - Top-down approach: similar to text grammars
    - Bottom-up approach: use of discourse relations
  - Interplay of content determination with discourse structuring

# Some problems of these standard approaches to discourse planning

- No theoretical backbone for content determination: What are the principles of content determination?

- No theory behind document structuring (RST is not a theory!)

- Document planning is an addressee-oriented process
  - Different users require diferrent content and different discourse structures

# Document planning: A game-theoretic approach

- A new approach: game-theoretic document planning (Klabunde 2009, Klabunde & Kornrumpf 2010)

- Aims of our work:

  - Treating document planning in NLG as interaction among independent agents (the system and a user model)

  - Using game-theoretic concepts for document planning

  - Establishment of rhetorical relations as speaker actions

  - Abductive reasoning for hearer actions

  - Nash-equilibria determine the optimal combinations of rhetorical relations with reasoning activities

# Document planning:  A game-theoretic approach

- Basic game-theoretic concepts:

- Game theory is the mathematical study of interaction between rational, self-interested agents

- Non-cooperative game theory is most interested in situations where agents' interests conflict, it's not restricted to these settings

  - the key is that the individual is the basic modeling unit, and that individuals pursue their own interests

  - cooperative/coalitional game theory has teams as the central unit, rather than agents

# Document planning:  A game-theoretic approach


A TCP backoff game:

Should you send your packets using correctly-implemented TCP (which has a "backoff" mechanism) or using a defective implementation (which doesn't)?

Consider this situation as a two-player game:

both use a correct implementation: both get 1 ms delay

one correct, one defective: 4 ms delay for correct, 0 ms for defective

both defective: both get a 3 ms delay.

# Document planning: Content determination and document structuring

- Finite, n-person game (N,A,u):
  - N is a finite set of *n* players, indexed by *i*
  - $A = A_1 \times \ldots \times A_n$, where $A_i$ is the action set for player *i*
    - $a \in A$ is an action profile, and so A is the space of action profiles
  - $u = (u_1, \ldots, u_n)$, a utility function for each player, where $u_i : A \rightarrow \mathbb{R}$

- Writing a 2-player game as a matrix:
  - row player is player 1, column player is player 2
  - rows are actions $a \in A_1$, columns are $a' \in A_2$
  - cells are outcomes, written as a tuple of utility values for each player

# Document planning: Content determination and document structuring

- Here's the TCP Backoff Game written as a matrix ("normal form"):

| | C | D |
|---|---|---|
| C | -1, -1 | -4, 0 |
| D | 0, -4 | -3, -3 |

# Document planning: Content determination and document structuring

- Prisoner's dilemma is any game:

| | C | D |
|---|---|---|
| C | a, a | b, c |
| D | b, c | d, d |

with c > a > d > b

# Document planning:  Content determination and document structuring

- Game-theoretic concepts:
  - How do we find the "best" action combinations for the players?
  - A Nash-equilibrium is a combination of actions in which no player has an incentive to change their action
  - Nash-equilibria are based on the notion of a best respond

# Document planning: Content determination and document structuring

- Game-theoretic concepts:

- If you knew what everyone else was going to do, it would be easy to pick your own action

- Let $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$
  - now $a = (a_{-i}, a_i)$

- Best response: $a^*_i \in BR(a_{-i})$ iff

$$\forall a_i \in A_i,\ u_i(a^*_i, a_{-i}) \geq u_i(a_i, a_{-i})$$

# Document planning: Content determination and document structuring

- Game-theoretic concepts:

- Now let's return to the setting where no agent knows anything about what the others will do

- What can we say about which actions will occur?

- Idea: look for stable action profiles.

- $a = a_1, \ldots, a_n$ is a ("pure strategy") Nash equilibrium iff

  $\forall i, a_i \in BR(a_{-i})$

# Document planning: Content determination and document structuring

- Nash equilibrium in the TCP Backoff Game:

| | C | D |
|---|---|---|
| C | -1, -1 | -4, 0 |
| D | 0, -4 | -3, -3 |

# Document planning: Content determination and document structuring

- Nash equilibrium in the TCP Backoff Game:

| | C | D |
|---|---|---|
| C | -1, -1 | -4, 0 |
| D | 0, -4 | -3, -3 |

# Document planning: Content determination and document structuring

- We want to use game-theoretic concepts for document planning, i.e., content selection and document structuring
- Two players: generation system $S$ and user model $L$
- $S$ can be thought of as the speaker
- $L$ simulates the behavior of the listener
- Rhetorical relations as speaker strategies

# Content determination

- Find a function *i* that maps every possible set of informational units to a set of messages that should be expressed in language:

  $i : P(D) \rightarrow P(M).$

- *D* is the set of data, *M the set of messages, and P* denotes the power set.

- In other words, *i* generates a set of messages from a given set $d \in P(D)$ and $d \subseteq D$, respectively.

## Definition: Content determination task

Given a set of information units $d \subseteq D$ and a mapping $i^* : d \rightarrow m \subseteq M$ which is known to be an optimal solution to the content determination problem by the speaker's experience, compute $m = i^*(d)$.

The content determination task is to transfer the knowledge source into a set of messages.

A caveat which remains is that we do not explicitly know $i^*$.

# Definition: Content determination task

Working hypothesis:

extend the definition of $i^*$ to $m = i^*(d) := h(g(d))$

where g : $d \rightarrow m^*$ is a function that generates a set of messages $m^*$ that can be derived from $d$ and

$h : m^* \rightarrow m$ is a function which maps messages to a complex message with respect to the preferences of $S$ and $L$.

In NLG terms, $g$ is responsible for content determination and $h$ determines the discourse relations.

# Listener action: abduction

- We use a domain theory to formulate the listener's knowledge base
- Knowledge and beliefs of Agent A about a specific domain D are called a domain theory $T_A(D)$ iff $T_A(D)$ is
  - a partially ordered set
  - a propositional logic program, i.e., each $t \in T_A(D)$ is of the form

    $\alpha_1 \wedge \ldots \wedge \alpha_i \wedge \neg\beta_1 \wedge \ldots \wedge \neg\beta_j \rightarrow p$ = body → head.

# Listener utility

- Definition (Listener utility):

- Let ($a_S = m$, $a_L = h$) be a strategy profile, $T_L$ the domain theory of $L$ and $\Psi(m)$ the propositions covered by m. Then the utility of the listener $L$ is defined as

  - $u_L(m, h) := \alpha_1 \cdot match_{T_L}(h, \Psi(m)) - \alpha_2 \cdot costs(h)$

    where $match_{T_L}(h, \Psi(m))$ measures how well the propositions in Ψ(m) are covered by $h$.

# Speaker utility

- Definition (Speaker utility)

- Be ($a_S = m$, $a_L = h$) be a strategy profile, $T_L$ the domain theory of $L$ and $complexity(m) = |nodes \in m|$ a metric for the complexity of $m$. Then the utility of S is defined as

$$u_S(m, h) := \frac{\beta 1 \cdot match_{T_L}(h,d)}{\beta 2 \cdot complexity(m)}$$

- The utility function for S has to find a balance between the goal of leading L to a hypothesis that accounts for d and minimizing the complexity of the communicated document plan m.

# A game-theoretic document planning algorithm

```
POOL ← all messages derived from d
N ← {S, L}
A_L ← H
u ← (u_S, u_S)
R ← nil
repeat
    A_S ← speaker actions: {rhet. relations which may link pairs of elements in POOL} ∪ POOL
    (m, h) ← pure strategy equilibrium of (N,A, u)
    if m ∈ POOL then
        R ← m
    else
        E ← {constituents of a_S}
        POOL ← POOL\E
        POOL ← POOL ∪ m
    end if
until R = nil
return R
```

## An example: the relevant parameters

- A runner wears a heart rate monitor

- Periodic measurements

- Goal: generate a document plan for the data

- In order to compute the discourse plan, the following needs to be specified:
  - The domain theory $T_L$
  - Document plans $\Psi(m)$ and a statement about their complexity (for production costs)
  - The user types
  - Assumption costs for hypotheses (costs(h))
  - Statements about relations between hypotheses and document plans ($match_{T_L}(h, \Phi(m))$)

# The document plans

| i | $\Phi(m_i)$ | *complexity*$(m_i)$ |
|---|---|---|
| 1 | hr_ high | 1 |
| 2 | hr_low | 1 |
| 3 | hr_high-then_low | 3 |

# User types

We define two user-types, the occasional runner *O* and the semi-professional runner *P*

| h | costs(h) | |
|---|---|---|
| ¬born_runner | 0 | 5 |
| ¬train regular | 1 | 4 |
| train_easy | 4 | 1 |
| ¬intensity full | 3 | 2 |
| intensity_to_high | 2 | 3 |
| state_good | 5 | 0 |

# User types and *costs(h)*

The following table specifies the values for $match_{TL}(h, \Phi(m))$:

| *h / Ψ(m)* | hr_high | hr_low | hr_high_then_low |
|---|---|---|---|
| ¬born_runner | 1 | -2 | -1 |
| ¬train_regular | 1 | -2 | -1 |
| train_easy | -2 | 1 | -1 |
| ¬intensity_full | -2 | 1 | -1 |
| intensity_too_high | 1 | 1 | 2 |
| state_good | -2 | -2 | -4 |

# P vs. S

| *h* / *Ψ(m)* | hr_high | hr_low | hr_high_then_low |
|---|---|---|---|
| ¬born_runner | (1, -1) | (-2, -1) | (-1, -0.33) |
| ¬train_regular | (0, -1) | (-3, -1) | (-2, -0.33) |
| train_easy | (-6, -1) | (-3, -1) | (-5, -0.33) |
| ¬intensity_full | (-5, -1) | (-2, -1) | (-4, -0.33) |
| intensity_too_high | (-1, 2) | (-1, -2) | (0, -0.67) |
| state_good | (-7, -4) | (-7, -4) | (-9, -1.33) |

# O vs. S

| *h* / *Ψ(m)* | hr_high | hr_low | hr_high_then_low |
|---|---|---|---|
| ¬born_runner | (-4,-1) | (-7,-1) | (-6,-0.33) |
| ¬train_regular | (-3,-1) | (-6,-1) | (-5,-0.33) |
| train_easy | (-3,-1) | (0,-1) | (-2,-0.33) |
| ¬intensity_full | (-4,-1) | (-1,-1) | (-3,-0.33) |
| intensity_too_high | **(-2,2)** | (-2,2) | (-1,-0.67) |
| state_good | (-2,-4) | (-2,-4) | (-4,-1.33) |

# Psycholinguistic aspects of document planning

- Document planning corresponds to certain aspects of conceptualization in language production:
  - Intention-based content determination
  - Document structuring is related to macroplanning
  - Addressee-orientation corresponds to the role of the common ground in language production
  - Furthermore, monitoring the conceptualization process is implicitely relevant for all interactive aspects of document planning

# Psycholinguistic aspects of document planning

- Comparison with conceptualization during language production
  - Levelt (1989, 1999): macro- and microplanning
    - Macroplanning: deciding which information to express and ordering this information
    - Microplanning: tailoring the information to language-specific demands
  - While macroplannning might be a universal process, microplanning is tailored to the requirements of the respective language
  - No sophisticated psycholinguistic model of macro- and microplanning available!
  - However: There is much work on language-specific conceptualization w.r.t. certain conceptual domains as, e.g., space and time

## Psycholinguistic aspects of document planning

- The role of monitoring: Postma (2000)

- Monitoring does only make sense in combination with feedback

- Errors at the conceptual level are followed by an appropriateness repair of the speaker

  *We start in the middle with – in the middle of the paper with a blue disc"*

  Appropriateness repair: reformulation (Blackmer & Mitton 1991)

- Errors in the formulation stage concern lexical selection, syntactic construction, or sound form encoding

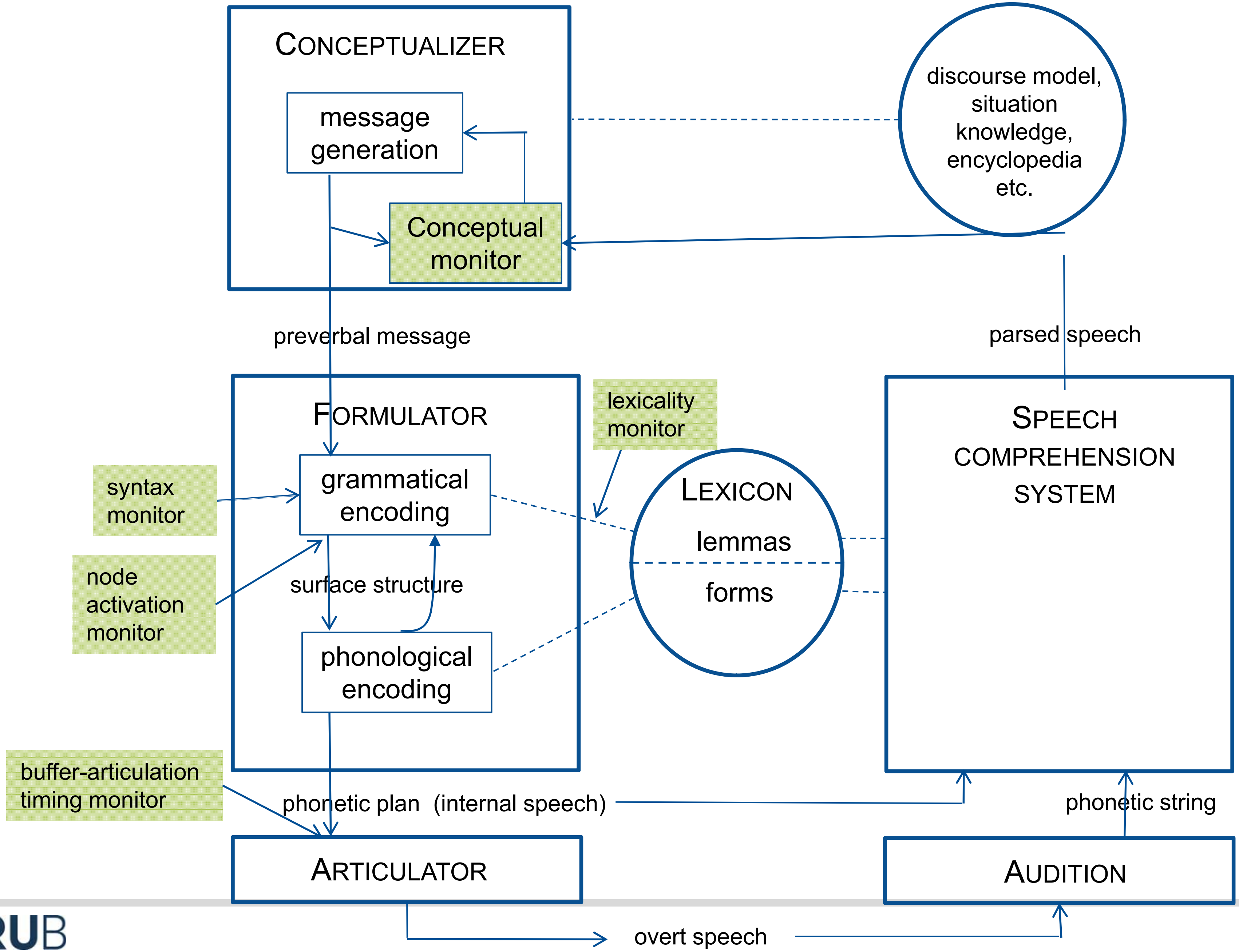
  *John comes – uh – like to come to the party*

  Syntactic deadlock: syntactic restructuring (Cutler 1983)

  *Left of purple is – uh – of white is purple*

  Lexical error: lemma substitution (De Smedt & Kempen 1987)

# Some proposed monitors in Levelt's model of language production



# Psycholinguistic aspects of document planning

- The conceptual loop is between the conceptualizer and the preverbal message
- Its function is appropriateness monitoring
- Since the conceptual loop is a kind of "thinking about one's thoughts", it is a metacognitive reflection (awareness of one's goals and intentions and those of others) (Frith, 1992)
- Conceptual errors  are repaired significantly slower than lexical or phonological inadequacies (Blackmer & Mitton 1991, Van Hest 1996).
- It seems that it is hard to reject a wrongly selected intention

# Psycholinguistic aspects of document planning

- How do monitor mechanisms work? Three approaches to speech monitoring have been proposed in the psycholinguistic literature:
  - Perceptual loop theory (Levelt 1983, 1989)
  - Production-based monitoring (Laver 1980, Schlenk et al. 1987)
  - Node structure theory (MacKay 1992)

# Psycholinguistic aspects of document planning

- The perceptual loop theory assumes that only certain end-products in the speech production are monitored
- These end-products are analysed in the same way as are utterances of others, i.e. with the speech comprehension system
- This theory postulates a single, central monitor within the conceptualizer
- It receives information from at least three channels:
  - The preverbal message (conceptual loop)
  - Information from the speech comprehension system
  - Communicative effects

# Psycholinguistic aspects of document planning

- Production-based monitoring focuses on the formulator
- In self-repairing, speakers have direct access to various processing components
- Components *inside* the formulator are accessible for monitoring
- Production monitors are special-purpose editors that form integral parts of the production system
- They comprise these channels:
  - Lexicality monitor
  - Syntax monitor
  - Node activation monitor
  - Plus different monitors for articulation processes

# Psycholinguistic aspects of document planning

- The node structure theory accounts for error detection in terms of the outflow of activation patterns in the node system

- Errors concern the activation of units that are novel at some level in the speech production hierarchy

- Since this theory has no say on conceptualization, we will not deal with this theory in detail (there is also strong empirical evidence against the NST)

# Psycholinguistic aspects of document planning

- Evaluation of the existing empirical support for the various accounts of speech monitoring
  (+: support, - and --: minor and major grounds for rejection, a blank: no convincing evidence available)

| | perception-based monitoring | production-based monitoring | Node structure theory | Comments |
|---|---|---|---|---|
| Location | + | | + | It might be argued that the complexity of error detection is best managed by a central, high-level error detection device. |
| Awareness | | + | | There are some indications for automatic, subconscious error detections both in speech production and other motor skills |
| Number of levels | + | - | -- | No difference in reported error rates between silent speech on the one hand and mouthed and noise-masked speech on the other hand suggests that there is no effective monitoring at the motor level. The best error detection occurs when auditory feedback is present, counter to NST. |

# Psycholinguistic aspects of document planning

- Lesson learned from speech monitoring models for NLG systems:
  - Monitoring and self-repair are genuine aspects of language production; reflected in the linguistic form of the speaker's utterance
  - Monitoring requires feedback, but feedback is not an issue in classical, pipelined NLG systems
  - Hence, if the wrong content has been selected, or if a defective part of a discourse plan has been created, there are no means to repair these plans immediately
  - Monitoring in NLG is an important topic in NLG

# Comparison with conceptualization during language production II

- The role of the common ground in speaking: Horton & Keyzar (1998)

- In most NLG systems, the common ground of system and user is treated in a simplified way: only the discourse history acts as common ground

- Human communication is based on a complex common ground, however: cultural and social knowledge, world knowledge, discourse history, meta-knowledge, believes, ...

- For example, referential descriptions (*the tiny cloud, the rainiest day,* ..) follow principles of "audience design" (Clark & Murphy 1982), especially the principle of optimal design:

> The speaker intends each addressee to base his inferences not on just *any* knowledge or beliefs he may have, but only on their *mutual* knowledge or beliefs – their common ground (Clark, 1992:81)

- Link to game theory!

# Comparison with conceptualization during language production II

- What role does the speaker's common ground play in the planning of utterances and the correction of errors, based on monitoring?

- Two models have been proposed in the psycholinguistic literature:
  - The initial design model:
    - Speakers apply the principle of optimal design in their utterances
  - The monitoring and adjustment model:
    - The common ground does not play a role in the initial plan of utterances.. Since speakers monitor their utterances, they may revise conceptual decisions to accommodate common ground

# Comparison with conceptualization during language production II

- The initial design model:
  - The utterance plan takes the addressee into account; it is tailored to the addressee's perspective
  - Only information that is part of the common ground is incorporated
  - Since the plan is tailored to the addressee already, little work for the monitoring process is left
  - The role of monitoring is only to detect errors that are inadequate w.r.t. the common ground

# Comparison with conceptualization during language production II

- The initial design model:

  - Example scenario:

    A bakery has two types of bread, both are round and one is 12cm in diameter and the other is 18cm in diameter. The speaker is interested in the 18cm-bread and says: "*I'd like a large loaf of bread, please*".

    The baker interprets the adjective *large* in the context of other breads which are present.

  - The contrast between the two diameters is common knowledge

  - *Because* the contrast set is part of the common ground, this information is used in planning the utterance

# Comparison with conceptualization during language production II

- The monitoring and adjustment model:
  - Speakers are egocentric; the utterance plan is not designed for the specific knowledge of the addressee
  - Consequently, the plan might occasionally rely on information whch is inaccessible to the addressee
  - Speakers monitor their productions, plans that do not sufficiently rely on non-shared information are revised to accomodate common ground
  - Hence, common ground operates as part  of a correction mechanism

# Comparison with conceptualization during language production II

- The monitoring and adjustment model:

  - Example scenario:

    A bakery has two types of bread, both are round and one is 12cm in diameter and the other is 18cm in diameter. The speaker is interested in the 18cm-bread and says: "*I'd like a large loaf of bread, please*".

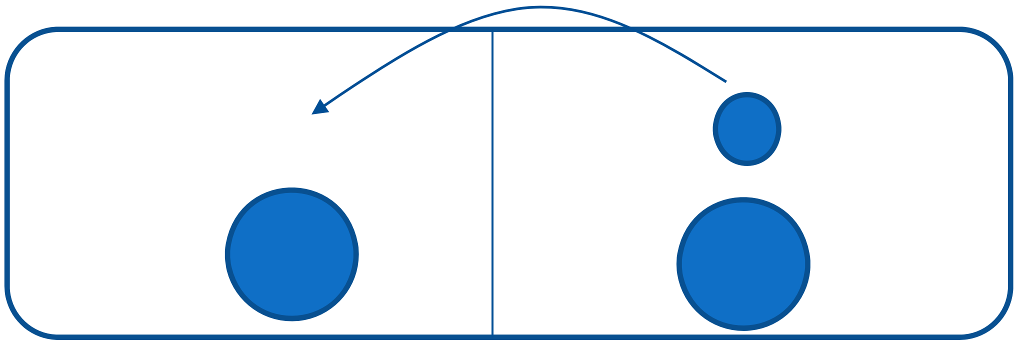
    The baker interprets the adjective *large* in the context of other breads which are present.

  - The referring expression had been used not because the contrast was common knowledge, but because it was salient

  - The utterance plan relied on the size contrast because it was present, *not* co-present

## Comparison with conceptualization during language production II

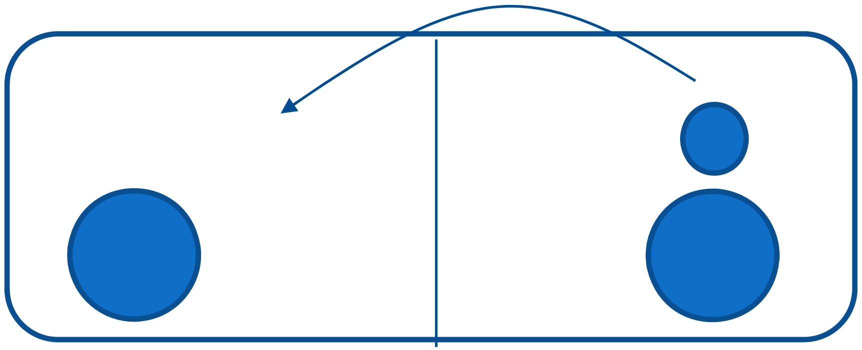
- A simple experiment to test the predictions of both models:
  - Subjects were speakers in a communication game
  - They saw one half of a computer screen
  - The addressee saw the other half of the screen
  - In each trial, two objects appeared on the speaker's side and then the top object moved to the addressee's side across a barrier:



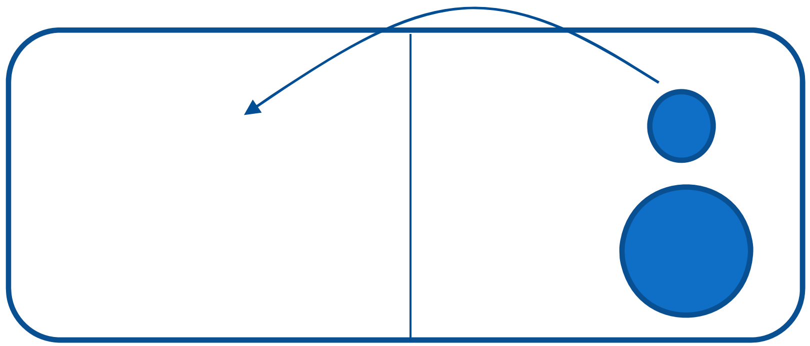
Listener | Speaker

# Comparison with conceptualization during language production II

- A simple experiment to test the predictions of both models:
  - The speaker's task was to describe the moving object so that the addressee could identify it
  - Distinguishing the accounts of both models by means of contrasting the shared context object with a case when this context is privileged to the speaker:



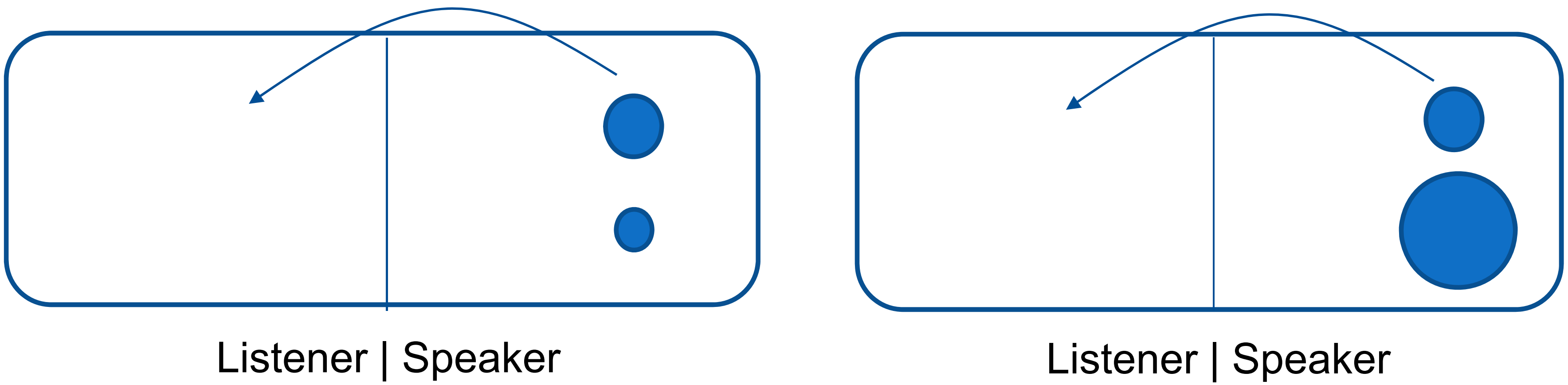
Listener | Speaker



Listener | Speaker

# Comparison with conceptualization during language production II

- A simple experiment to test the predictions of both models:
  - Initial design model: speakers use the context when it is shared but not when it is privileged
  - Monitoring and adjustment model: the plan uses the context information regardless of whether shared or privileged
  - Changing the context in the privileged scenario: does the description change?
  - In an additional study, the same tasks had to be performed with the speakers put under time pressure



# Document planning: Content determination and document structuring

- A simple experiment to test the predictions of both models:
- Results:
  - Speaker's descriptions relied on privileged context information less than on shared context information
  - ... but when speeded they relied on shared and privileged context to the same degree
  - Utterance planning seems to be agnostic w.r.t. common ground information
  - There is some evidence against the initial design model, but time pressure increases the speaker's willingness to take the common ground into account

## Psycholinguistic aspects of document planning

- Lesson learned from common ground models for NLG systems:
  - The common ground is not just the discourse history; we need non-linguistic context information
  - Considering the common ground (for referential descriptions) is a dynamic process that seems to depend on a number of factors
  - The common ground plays an important role in the generation of referring expressions (the topic of our next session)

# Outline of this course