

# Building Applied Natural Language Generation Systems

Robert Dale and Ehud Reiter

# Overview

- 1 An Introduction to NLG
- 2 Requirements Analysis for NLG
- 3 NLG Architecture and System Design
- 4 A Case Study
- 5 A Closer Look at the Component Tasks
- 6 Conclusions and Pointers

# Component Tasks in NLG

- Text Planning
  - Content determination
  - Discourse planning
- Sentence Planning
  - Sentence aggregation
  - Lexicalisation
  - Referring expression generation
- Linguistic Realization
  - Syntactic and morphological realization
  - Orthographic realization

# Text Planning

## Goal:

- To determine what messages to communicate, and how to rhetorically structure these messages

## Approaches

- Methods based on reasoning about intention
- Methods based on observations about text structure
- More general expert system-based methods

# Methods based on Reasoning about Intentions

Typically use AI planning techniques:

- Goal = desired communicative effect
- Plan constituents = things the system can say (speech acts)
- involves reasoning about user's beliefs to determine what to say and how to rhetorically structure that information
- requires reasoning about why information should be included

# Methods based on Reasoning about Intentions

Analysis:

- principled and theoretically elegant
- expensive and complex
- may be most appropriate for dialog systems
- not widely used in applied NLG systems

# Methods based on Text Structure

Basic idea:

- texts often follow conventionalised patterns
- these patterns can be captured by means of 'text grammars' that dictate both content and coherent structure
- these effectively recompile some of the reasoning that has to be carried out explicitly in more sophisticated methods

# Methods based on Text Structure

Text formats encapsulated in SCHEMAS:

- template programs which produce text plans
- specify how a particular text plan can be constructed using smaller schemas or atomic messages
- can specify many degrees of variability and optionality
- may include content determination or only text structuring



# Methods based on Text Structure

Implementing schemas:

- simple schemas can be expressed as grammars
- more flexible schemas usually implemented as macros or class libraries on top of a conventional programming language, where each schema is a procedure
- usually embody ideas from Rhetorical Structure Theory to provide text coherence
- currently most popular text-planning approach in applied NLG

# An Example Schema

Schema-name: PreviousAttemptFailedParagraph()

condition: PreviousAttemptToQuit == TRUE

body:

N1 = SummarisePreviousAttempts();

N2 = LookForEncouragingSigns();

N3 = Message("Most people who successfully quit smoking make several unsuccessful attempts first ...")

If (N2 == NULL)

return Contrast(N1, N3)

else return Contrast(N1, Elaboration(N2, N3))

# Deriving Schemas from a Corpus

## Using the Target Text Corpus:

- take a small number of similar corpus texts
- break up these texts into messages, and try to determine how each message can be computed from the input data
- propose rules or structures which explain why message  $x$  is in text A but not text B -- this may be easier if messages are organised into a taxonomy
- discuss this analysis with domain experts, and iterate
- repeat the exercise with a larger set of corpus texts

# Schemas: Pros and Cons

## Advantages of schemas:

- Computationally efficient
- Allow arbitrary computation when necessary
- Naturally support genre conventions
- Relatively easy to acquire from a corpus

## Disadvantages

- Less flexible
- May be difficult to reuse in other applications

# Expert System Approaches

Use (constructive) expert-system techniques

- production rules
- case-based reasoning

Analysis

- Places planning and schemas into perspective
- Suggests many other possible algorithms

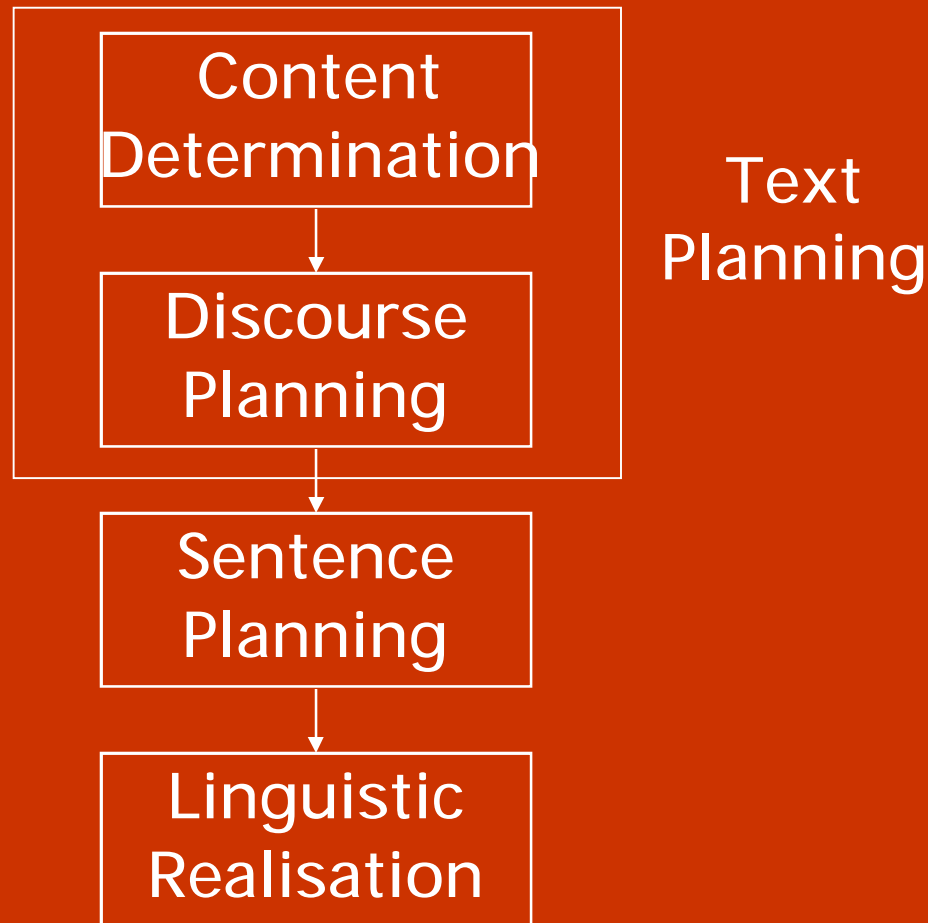
# A Production Rule Example

```
if (and (?deposit < ?total_cost)
        (case :disputed-charges))
then
  (add-node
   (rhetorical fact-explanation
    (message ?service_provider has submitted an
             additional charge to your account)
    (message ?deposit was less than
             ?total_cost)))
```

# Research Issues

- Using other ideas from expert systems -- eg case based reasoning
- Principled ways of integrating shallow and deep reasoning
- A better understanding of rhetorical relations

# The WeatherReporter Architecture





# Text Planning

Content Determination:

- Build messages for the reporting period

Discourse Planning

- Organize messages to be presented in the text

# Content Determination

## Determining Message Types:

- Break texts into largest possible base informational elements
- Identify underlying data required for each kind of informational element
- define a set of message types that collect together underlying data in ways that are convenient for linguistic expression

# Content Determination

The Key Issues in Message Specification:

- determine the most useful clusterings of information
- represent these clusterings in a way that makes manipulation for linguistic expression easy

# Content Determination

- Routine messages
  - MonthlyRainFallMsg,  
MonthlyTemperatureMsg,  
RainSoFarMsg,  
MonthlyRainyDaysMsg
- Always constructed for any summary to be generated

# Content Determination

- Significant Event messages
  - RainEventMsg,  
RainSpellMsg,  
TemperatureEventMsg,  
TemperatureSpellMsg
- Only constructed if the data warrants their construction: eg if rain occurs on more than a specified number of days in a row

# Content Determination

## A MonthlyRainfallMsg:

```
((message-id msg091)
 (message-type monthlyrainfall)
 (period ((month 04)
           (year 1996)))
 (absolute-or-relative relative-to-average)
 (relative-difference ((magnitude ((unit millimeters)
                                   (number 4)))
                       (direction +)))))
```

# Content Determination

## A RainSpellMsg:

```
((message-id msg096)
 (message-type rainspellmsg)
 (period ((begin ((day 04)
                  (month 02)
                  (year 1995)))
          (end ((day 11)
                (month 02)
                (year 1995)))
          (duration ((unit day)
                    (number 8)))))
 (amount ((unit millimetres)
          (number 120))))
```

# Discourse Planning

- The content of the text to be generated has been determined by the message building phase
- The goal of discourse planning is to assemble this set of messages into a coherent text



# Discourse Planning

A Simple Discourse Schema:

WeatherSummary →

Monthl yTempMsg

Monthl yRai nfal l Msg

Rai nyDaysMsg

Rai nSoFarMsg

# Discourse Planning

Limitations of a simple schema-based approach:

- Very rigid
- Doesn't allow for special circumstances

Ways of adding more flexibility:

- add optionality to the schema to deal with special cases
- segment knowledge of what makes a text coherent into separate rules

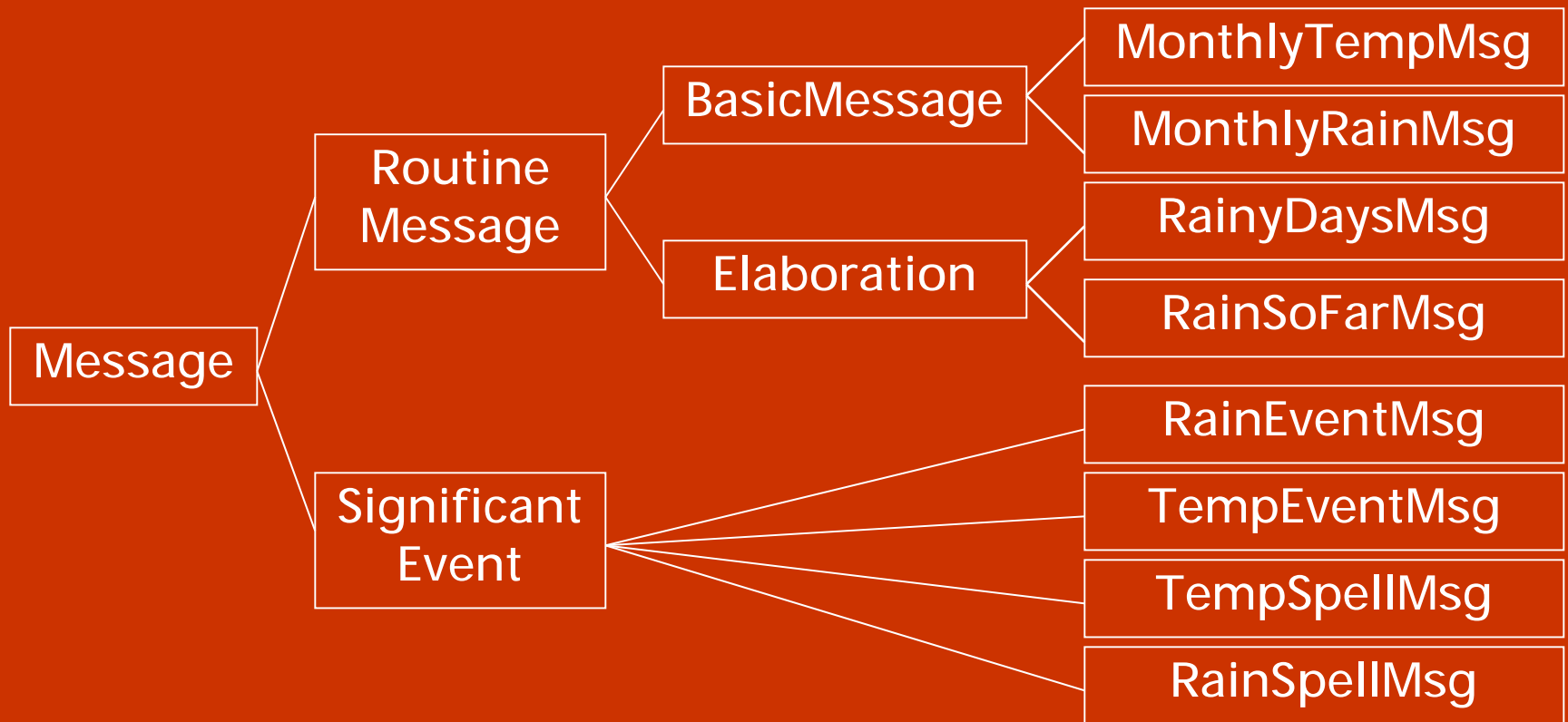
# Discourse Planning

Three basic rhetorical relationships:

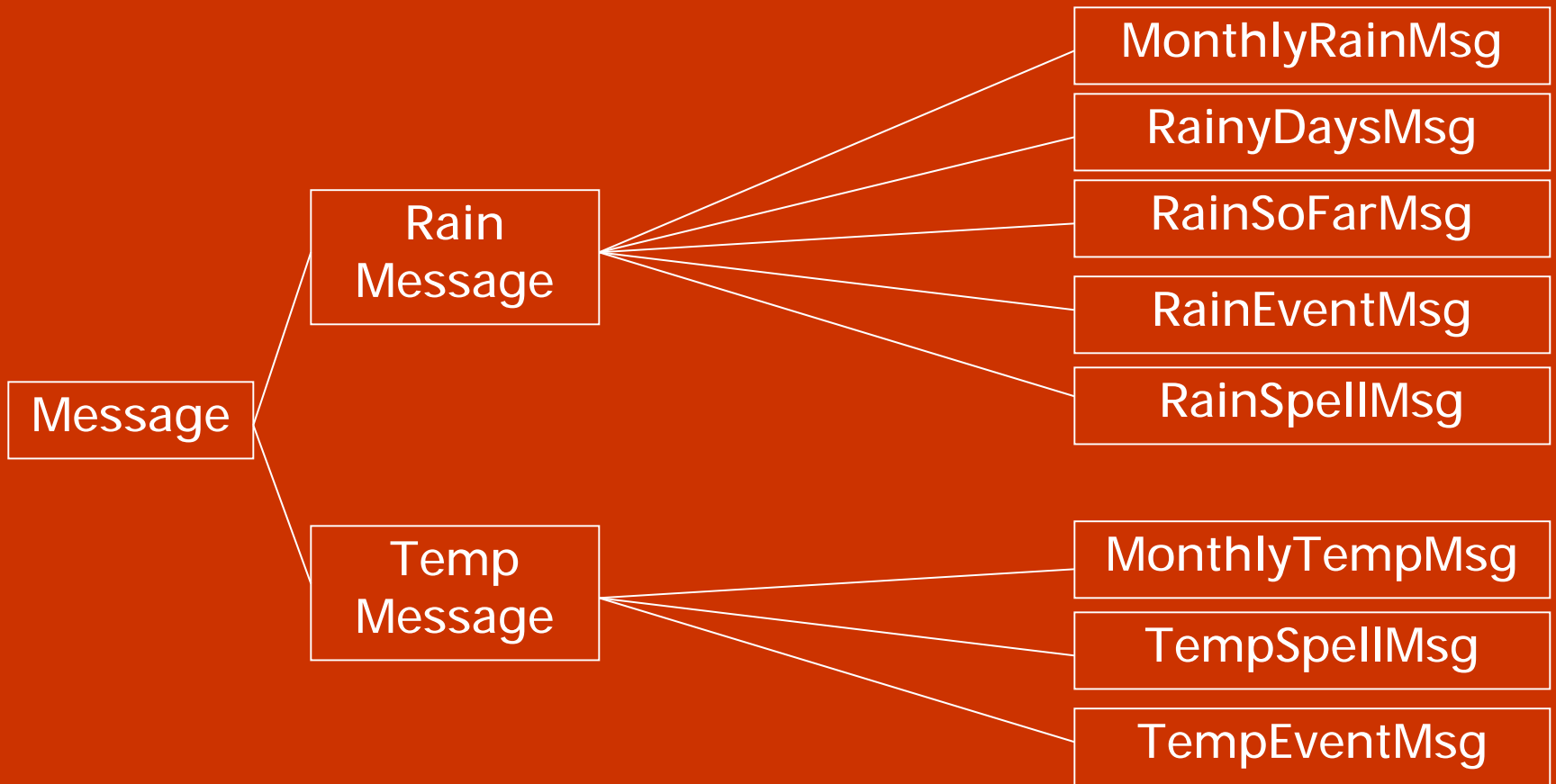
- SEQUENCE
- ELABORATION
- CONTRAST

Applicability of rhetorically-based planning operators determined by Message Status and Message Topic

# Message Status



# Message Topic



# Discourse Planning

## SEQUENCE

- Two messages can be connected by a SEQUENCE relationship if they are both BasicMessages

# Discourse Planning

## ELABORATION

- Two messages can be connected by an ELABORATION relationship if:
  - they are both have the same TOPIC
  - the nucleus is a BasicMessage

# Discourse Planning

## CONTRAST

- Two messages can be connected by a CONTRAST relationship if:
  - they both have the same TYPE
  - they both have the feature  
absol ute-or-rel ati ve = rel ati ve-to-average
  - they have different values for  
rel ati ve-di fference: di recti on



# Discourse Planning

- Select a start message
- Use rhetorical relation operators to add messages to this structure until all messages are consumed or no more operators apply
- Start message is any routine message

# Discourse Planning

The algorithm:

TextPlan = StartMessage

MessageSet = MessageSet - StartMessage

repeat

- find a rhetorical operator that will allow attachment of a message to the TextPlan
- attach message and remove from MessageSet

until MessageSet = 0 or no operators apply

# Target Text #1

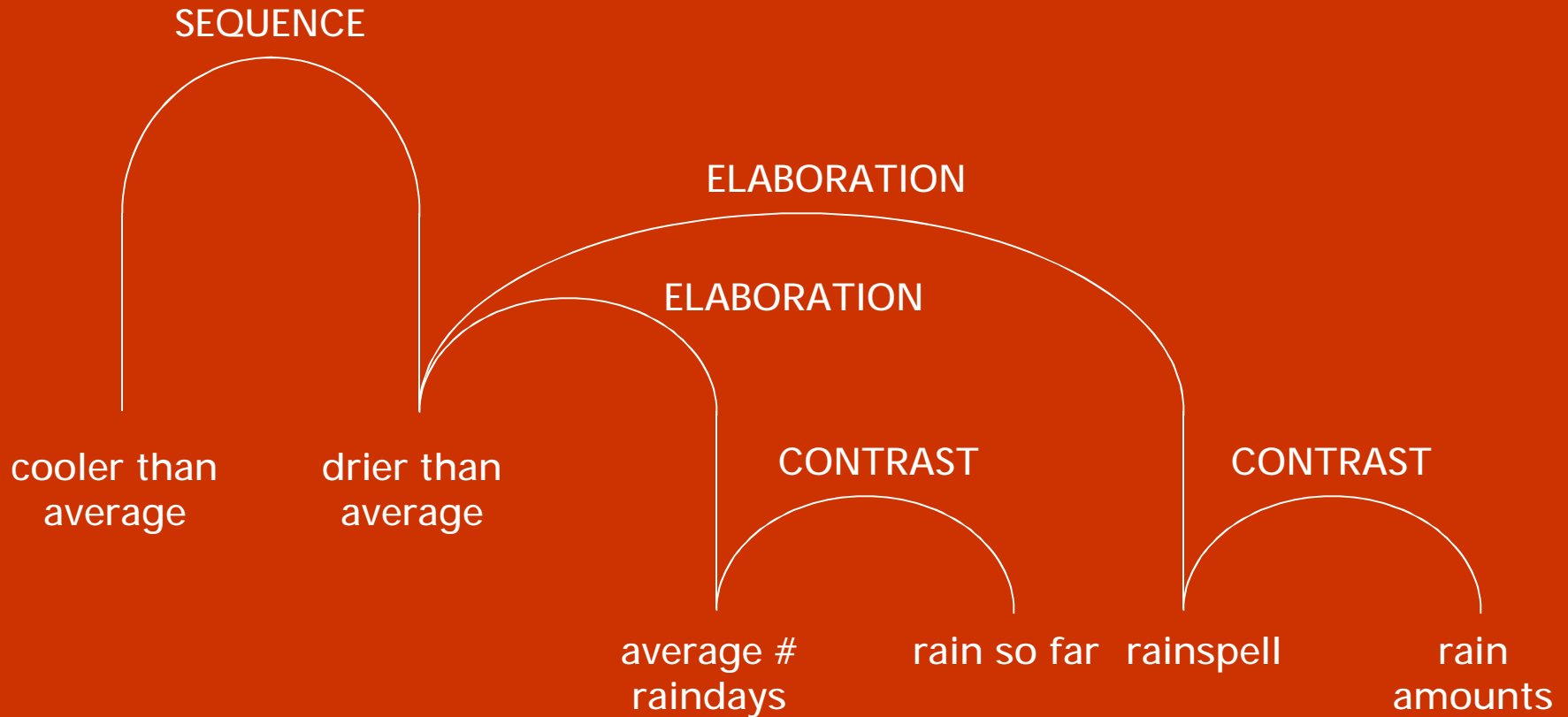
The month was cooler and drier than average, with the average number of rain days, but the total rain for the year so far is well below average. Although there was rain on every day for 8 days from 11th to 18th, rainfall amounts were mostly small.

# Discourse Planning

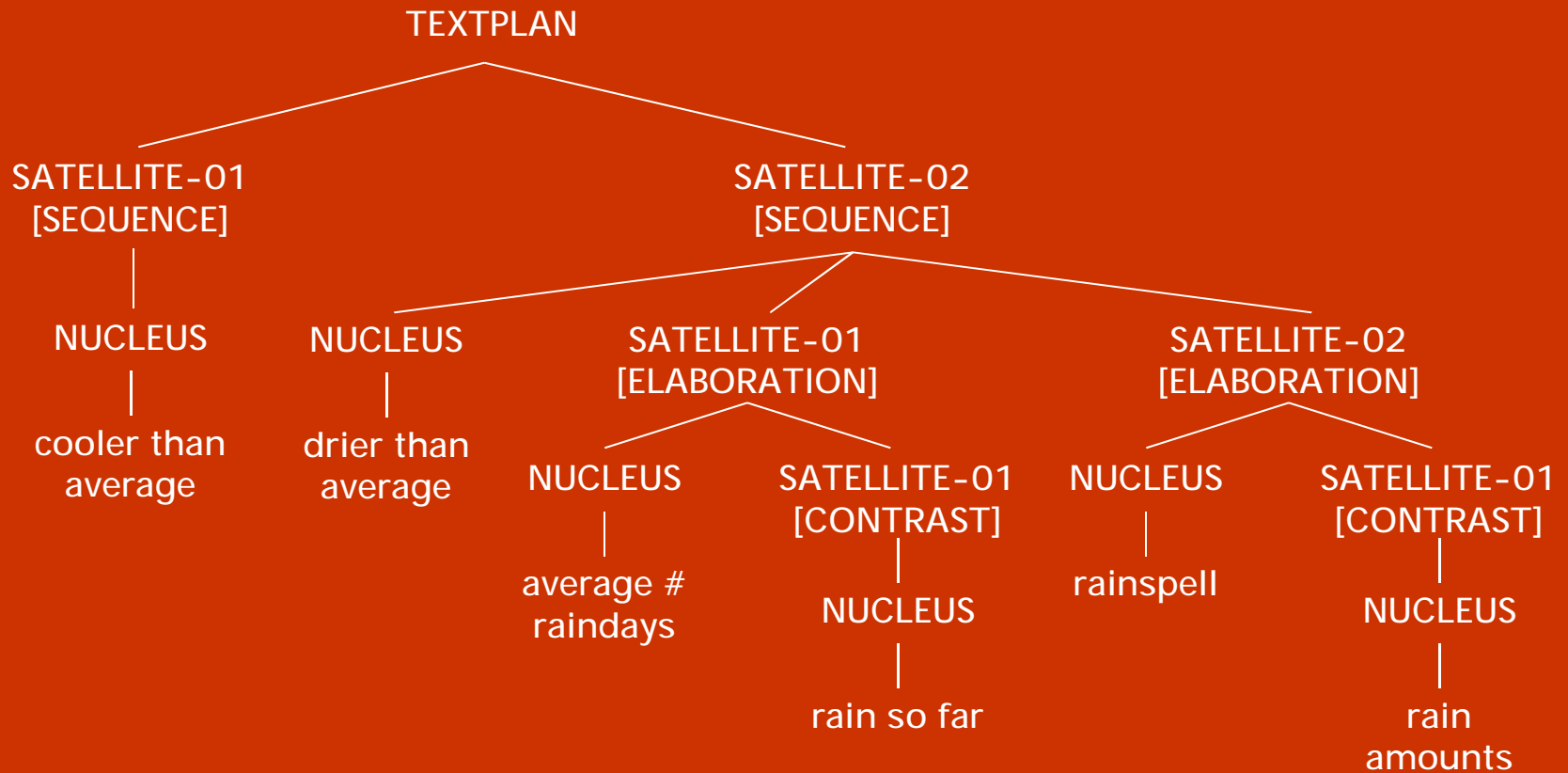
The Message Set:

CoolerThanAverage  
DrierThanAverage  
AverageNumberOfRainDays  
RainSoFar  
RainSpell  
RainAmounts

# Discourse Planning



# A Text Plan



# A Text Plan

```
((type textplan)
 (relations ((sequence satellite-01 satellite02)))
 (satellite-01 ((nucleus <MONTHLYTEMPMSG>)))
 (satellite-02 ((nucleus <MONTHLYRAINFALLMSG>
   (relations ((elaboration nucleus satellite-01)
     (elaboration nucleus satellite-02)))
   (satellite-01 ((relations ((contrast nucleus satellite-01)
     (nucleus <RAINYDAYSMMSG>)
     (satellite-01 ((nucleus <RAINSOFARMSG>))))))
   (satellite-02 ((relations ((contrast nucleus satellite-01)
     (nucleus <RAINSPELLMSG>)
     (satellite-01 ((nucleus <RAINAMOUNTSMMSG>))))))))))
```

# Discourse Planning

- Result is a TEXT PLAN: a tree structure populated by messages at its leaf nodes
- Next step: realising the messages as text



# A Simple Realiser

- We can produce one output sentence per message in the discourse plan
- A specialist fragment of code for each message type determines how that message type is realised

# A Simple Realiser

For the MonthlyTemperatureMsg:

```
TempString = case (TEMP - AVERAGETEMP)
  [2.0 ... 2.9]:  ' very much warmer than average.'
  [1.0 ... 1.9]:  ' much warmer than average.'
  [0.1 ... 0.9]:  ' slightly warmer than average.'
  [-0.1 ... -0.9]: ' slightly cooler than average.'
  [-1.0 ... -1.9]: ' much cooler than average.'
  [-2.0 ... -2.9]: ' very much cooler than average.'
endcase
Sentence = 'The month was' + TempString
```

# One Message per Sentence

- 1 The month was cooler than average.
- 2 The month was drier than average.
- 3 There were the average number of rain days.
- 4 The total rain for the year so far is well below average.
- 5 There was rain on every day for 8 days from 11th to 18th.
- 6 Rainfall amounts were mostly small.

# Target Text #1

The month was cooler and drier than average, with the average number of rain days, but the total rain for the year so far is well below average. Although there was rain on every day for 8 days from 11th to 18th, rainfall amounts were mostly small.

# Simple Templates

Problems with simple templates:

- MonthlyTemp and MonthlyRainfall don't always appear in the same sentence
- When they do appear in the same sentence, they don't always appear in the same order
- Each can be realised in different ways: eg 'very warm' vs 'warmer than average'
- Additional information may or may not be incorporated into the same sentence

# Component Tasks in NLG

- Text Planning
  - Content determination
  - Discourse planning
- Sentence Planning
  - Sentence aggregation
  - Lexicalisation
  - Referring expression generation
- Linguistic Realization
  - Syntactic and morphological realization
  - Orthographic realization