Textual Representation of
Plan Explanations
Contents

- Fundamentals
- Generating Textual Explanations
- Demonstration
1. Fundamentals

- Hybrid Plans
- Formal Explanations
- Natural Language Generation
1.1. Hybrid Plans

Diagram:

- **enterAlbum**
- **selectPic**
- **pressSendByEMail**
- **chooseRecipient**
- **pressSend**
- **displayPic**
- **sendPic**
- **top**

Events:

- **InAlbumMode**
- **PicSelected**
- **PicAttached**
- **RecipientSet**
- **EMailSent**
- **goal**

Transitions:

- **dPViaAlbum**
- **mtop**

Flow:

1. **enterAlbum** → **InAlbumMode** → **selectPic** → **PicSelected**
2. **PicSelected** → **pressSendByEMail** → **PicAttached** → **chooseRecipient**
3. **chooseRecipient** → **pressSend** → **EMailSent** → **goal**
4. **pressSend** → **sendPic** → **sendPicByEMail** → **top**
1.2. Formal Explanations

Proving a task's necessity = Explaining a task's necessity
1.2. Formal Explanations

Three relations:

- Necessary(t)
- CausalRelation(t1,φ,t2)
- DecompositionRelation(t1,m,t2)
1.2. Formal Explanations

C-CH:

\[ \forall t : \text{Task} \ . \ [\text{Necessary} \ (t) \ \leftarrow \ \exists t' : \text{Task} \ ; \ \phi : \text{Formula} \ . \ [\text{CausalRelation}(t, \phi, t') \ \land \text{Necessary} \ (t')] \]
1.2. Formal Explanations

D-CH:

\[ \forall t: \text{Task} . \ [\text{Necessary} \ (t) \leftarrow \exists t': \text{Task} ; m: \text{Method} . \ [\text{DecompositionRelation}(t,m,t') \land \text{Necessary} \ (t')] ] \]
## 1.2. Formal Explanations

1. Necessary(press_newAppointment)  
   - C-Ch 4,5
2. CausalRelation(press_newAppointment, \( \phi \), set_Name)  
   - Basic
3. DecompositionRelation(set_Name, \( m \), configure_Appointment)  
   - Basic
4. CausalRelation(press_newAppointment, \( \phi \), configure_Appointment)  
   - C-Prop 2,3
5. Necessary(configure_Appointment)  
   - C-Ch 8,9
6. CausalRelation(set_Time, isSet_Time\_Appointment,Date, press_OK)  
   - Basic
7. DecompositionRelation(set_Time, \( m \), configure_Appointment)  
   - Basic
8. CausalRelation(configure_Appointment, isSet_Time\_Appointment,Date, press_OK)  
   - P-Prop 6,7
9. Necessary(press_OK)  
   - C-Ch 10,11
10. CausalRelation(press_OK, created\_Appointment, goalTask)  
    - Basic
11. Necessary(goalTask)  
    - Basic
1.3. Natural Language Generation

- Goal: natural and easy to read Text
- Six characteristic phases
1.3. Natural Language Generation

- Text Planning:
  - Content Determination
  - Discourse Planning

- Sentence Aggregation
1.3. Natural Language Generation

- Lexicalization
- Referring Expression Generation
- Linguistic Realization
2. Generating Textual Explanations

- Goals
- Text Planning
- Sentence Aggregation
- Lexicalization
- Referring Expression Generation
- Linguistic Realization
2.1. Goals

Goal:
A system that generates an easy to read, variety rich textual explanation in the english language from a formal explanation.
2.1. Goals

Suggestion:

1. Necessary(press_newAppointment)
2. CausalRelation(press_newAppointment, φ, set_Name)
3. DecompositionRelation(set_Name, m, configure_Appointment)
4. CausalRelation(press_newAppointment, φ, configure_Appointment)
5. Necessary(configure_Appointment)
6. CausalRelation(set_Time, setIs_TimeAppointment_Date, press_OK)
7. DecompositionRelation(set_Time, m, configure_Appointment)
8. CausalRelation(configure_Appointment, setIs_TimeAppointment_Date, press_OK)
9. Necessary(press_OK)
10. CausalRelation(press_OK, createdAppointment, goalTask)
11. Necessary(goalTask)

„press_newAppointment is necessary since this task is necessary for configure_Appointment, that one is needed for press_OK and that one is necessary for establishing the goal.“
2.1. Goals

- More appropriate expressions for tasks etc.
- Involving relations
- Adjustable Lexicalization
- Fast adjustability to switched domains etc.
- Preparing all necessary input files for the Smartphone-Domain
2.2. Text Planning

- Input: formal explanation
- Output: Text Tree
2.2. Text Planning

Types of inner nodes:
  - Sequence (STOP)
  - Sequence (COMMA)
  - Elaboration
  - Message
2.2. Text Planning

Types of messages (leaves):

- Necessary
- Decomp
- Causal
- Phi
- Goal
- Top
2.2. Text Planning

Two variations of Text Planning

- **Simple**: Constructs a Simple Text Tree
- **Advanced**: Uses Phi Nodes
2.2. Text Planning

1. Necessary(t1)  
2. DecompositionRelation(t1, m, t2)  
3. Necessary(t2)  
4. CausalRelation(t3, \( \phi_3 \), t4)  
5. DecompositionRelation(t3, m, t2)  
6. CausalRelation(t2, \( \phi_4 \), t4)  
7. Necessary(t4)  
8. CausalRelation(t4, \( \phi_5 \), goalTask)  
9. Necessary(goalTask)
2.2. Text Planning

SEQUENCE

ELABORATION

necessary(t1) decomp(t1,t2)

ELABORATION

necessary(t2) causal(t2,t4)

ELABORATION

necessary(t4) goal(t4)
2.3. Sentence Aggregation

- Input: Text Tree

- Output: Sentence Tree

- delete redundant information, shorten the textual explanation and simplify the sentence structure
2.3. Sentence Aggregation

- Sentences can contain more than one elaboration node.

- Sequence Nodes (COMMA) can be used to easily concatenate elaboration nodes.

- Necessary Messages from inner elaboration nodes can be deleted since their information is already expressed through the context.
2.3. Sentence Aggregation

• Three Variations:
  
  - Simple: Does not change the tree
  - Content-based: unites subtrees with similar messages
  - Length-based: unites subtrees to sentences with randomly chosen length
2.3. Sentence Aggregation

[Diagram showing a hierarchy of sentence structures with labels such as "SEQUENCE(STOP)", "ELABORATION", "necessary(t1)", "causal(t1,t2)", etc.]
2.4. Lexicalization

- Input: Sentence Tree
- Output: lexicalized Sentence Tree
2.4. Lexicalization

- Choose for each message a predefined sentence fragment

- Inserting task names, arguments, etc.

- Three variations
2.4. Lexicalization

Simple Lexicalization:

- task names remain unchanged
- arguments of tasks can not be included
- resulting text is similar to the example
2.4. Lexicalization

Advanced Lexicalization:

- requires additional input
- can express tasks and their arguments in a seemingly natural way
2.4. Lexicalization

Input Supporting Lexicalization:

- offered sentence fragments can be modified
- other functions identical to the Advanced Lexicalization
2.4. Lexicalization

Example (Sentence Fragment):

;is part of;<=>0,i,to;1,c,;<=>0;0;-1;-1;1;1;

means:

"to" + <task No. 0 in infinitive form> + "is part of" + <task No. 1 in continuous form>
2.4. Lexicalization

Example (Task):

Task:
enter_Number_ForSMS <=>
"enter number" <=>
enter %1%'s number <=>
entering %1%'s number
2.5. Referring Expression Generation

Produced text without R.E.G.:

Pressing the [eMail] button is crucial since to press the [eMail] button is part of entering the eMail menu, the task "enter the eMail menu" is needed for pressing [New eMail] and the task "press [New eMail]" is necessary as pressing [New eMail] is needed for pressing [Send]. To press [Send] is crucial since it establishes a goal.
2.5. Referring Expression Generation

- 'it', 'that' and 'which' are supported

- Precursing sentence fragment offers tasks to be replaced in the following sentence

- Following sentence accepts the offered task at specified locations when they match its own task at that position
2.5. Referring Expression Generation

Example:

Necessary Message:

;is needed;<=>0,c,;<=>0;0;0;0;-1;-1

Causal Message:

;is needed for;<=>0,i,to;1,n,the task;<=>0;0;-1;-1;1;1

=> task 0 will be expressed through 'it'
2.6. Linguistic Realization

- Reducing the tree to a string
- Punctuation, whitespaces
- Expressions of inner nodes
3. Demonstration

<table>
<thead>
<tr>
<th></th>
<th>Necessary(pressHome.EMail)</th>
<th>D-Ch 2,3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>DecompositionRelation(pressHome.EMail, m, enterMode.EMail)</td>
<td>[*] C-CH 4,5</td>
</tr>
<tr>
<td>3</td>
<td>Necessary(enterMode.EMail)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CausalRelation(enterMode.EMail, inMode_EMail, pressEMail.NewEMail)</td>
<td>[*] C-Ch 6,7</td>
</tr>
<tr>
<td>5</td>
<td>Necessary(pressEMail.NewEMail)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CausalRelation(pressEMail.NewEMail, φ₁, pressEMail.NewEMail.Send)</td>
<td>[*] C-Ch 8,9</td>
</tr>
<tr>
<td>7</td>
<td>Necessary(pressEMail.NewEMail.Send)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CausalRelation(pressEMail.NewEMail.Send, φ₂, goalTask)</td>
<td>Basic</td>
</tr>
<tr>
<td>9</td>
<td>Necessary(goalTask)</td>
<td></td>
</tr>
</tbody>
</table>
3. Demonstration

- Simplest:

The task press_Home.EMail is necessary because it belongs to the task enterMode_EMail. That is obligatory since it is needed for the task press_EMail.NewEMail and that is crucial as it is needed for the task press_EMail.NewEMail.Send. That is necessary since it establishes a goal.
3. Demonstration

- Most advanced:

Pressing the [eMail] button is crucial since it is part of the task “enter the eMail menu”, which is needed for pressing [New eMail] and that is necessary as it is needed for pressing [Send]. That is crucial since it establishes a goal.