

# ISO/DIT dialogue annotation and its semantics



Harry Bunt



OTIM-ILIKS Workshop, Aix-en-Provence, May 2011

# Plan

- 1 Introduction
- 2 Multidimensionality and dimensions
- 3 DIT<sup>++</sup>, and ISO 24617-2
- 4 Annotation scheme design methodology → Abstract and concrete syntax *and semantics* of annotations
- 5 DiAML (Dialogue Act Markup Language)
- 6 DiAML semantics
- 7 Concluding remarks; Perspectives

1.

# Introduction

# Semantic approach to dialogue acts

Dialogue acts are viewed semantically as *information-state update operators* (or 'context update' operators)

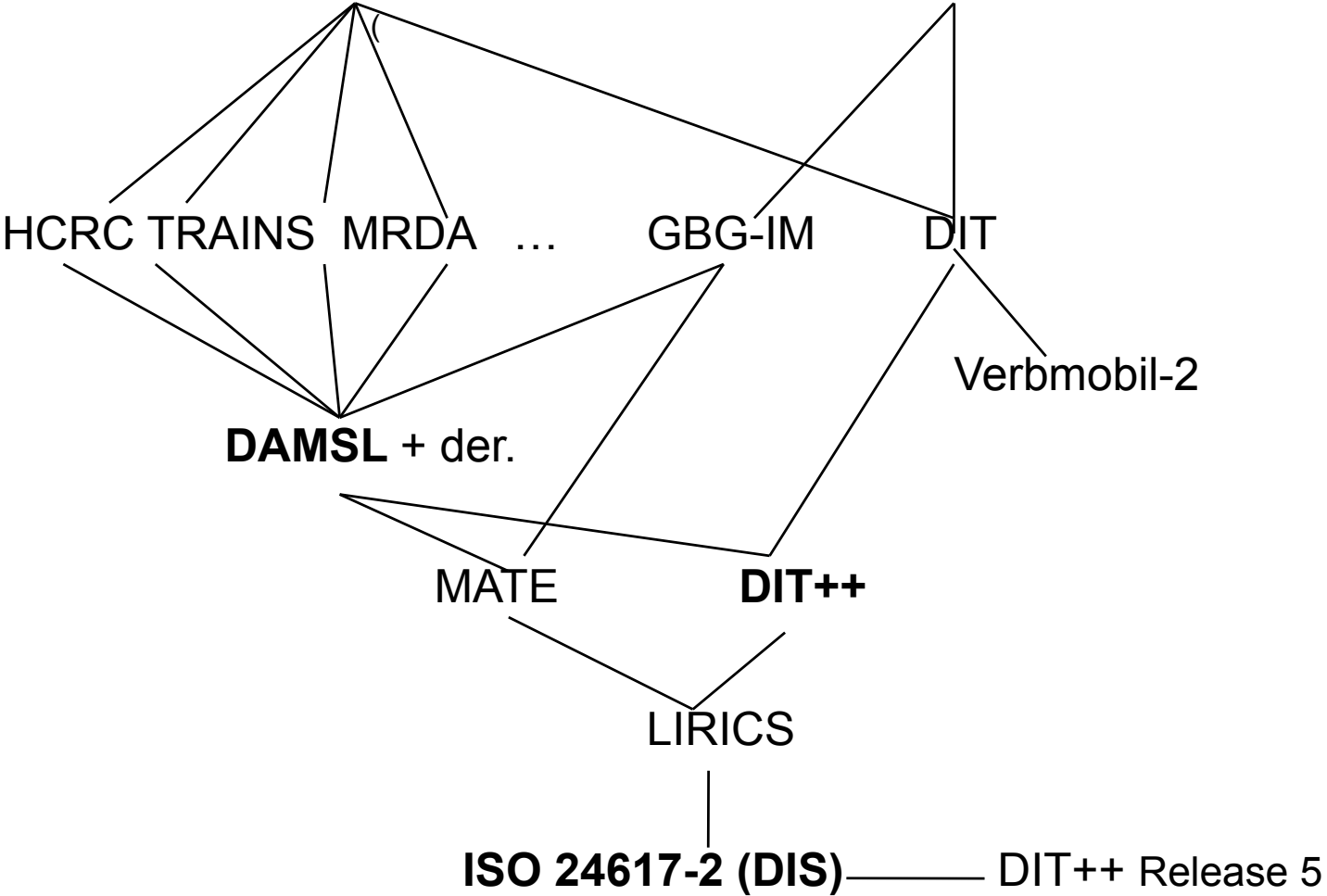
***Communicative function*** and ***semantic content*** ;

- the communicative function specifies a type of update operation;
- the semantic content is the material for updating the IS with.

# Dialogue act analysis frameworks

Speech Act Theory  
(Austin, Searle)

Communication as Cooperation (Grice)  
Communicative Activity Analysis (Allwood)



2.

# Multidimensionality and dimensions

# Multidimensionality

→ Utterances in dialogue often may more than one communicative function – be ***multifunctional***

Participating in a dialogue involves more than ‘just’ pursuing a certain goal, task or activity:

- Giving and eliciting feedback
- Taking turns
- Managing the use of time
- Establishing and maintaining contact
- Dealing with social obligations (greeting, thanking, apologizing, ...)
- .....

→ Communication has many ‘***dimensions***’

→ An utterance may have a function in more than one dimension

# Dimensions for dialogue acts

Definition:

*A multidimensional taxonomy of dialogue acts* is a set of dialogue acts, partitioned into clusters of which each member addresses the same aspect of communication, and where each of these aspects can be addressed independently of the other dimensions.

(In other words, dimensions are '*independent*' or '*orthogonal*'.)

DIT: Every two communicative functions which can be used in a certain dimension are either mutually exclusive or one is a specialization of the other (e.g. *Confirmation* → *Answer*; *Correction* → *Disagreement*)

→ a functional segment can have a communicative function in each dimension, but never more than one (*modulo entailed functions*)



# DIT: 10 dimensions of communication

- **Task:** Performing a certain '*task*' or '*activity*' through the dialogue
  - **Feedback:**
    - auto-feedback;** providing information about one's *processing* (perception, understanding, evaluation,...) of previous utterances;
    - allo-feedback:** asserting or eliciting information the *partner's processing* of previous utterances
  - **Interaction Management: managing**
    - **contact**
    - **turn** allocation (speaker role)
    - use of **time**
    - **structuring** of the **discourse**
    - **editing** of **one's own** and of **partner's speech**
    - **social obligations** - *greeting, thanking, apologising, saying goodbye,...*
- *All these types of information must be represented in information states for ISU semantics of dialogue acts !*

# What's NOT a dimension...

DAMSL dimensions include:

- *'Info-Request'*
- *'Signal Understanding'* (feedback)

S: Did you say Thursday?

Which dimension?

# What's NOT a dimension...

DAMSL dimensions include:

- *'Info-Request'*;
- *'Signal Understanding'* (feedback)

S: Did you say Thursday?

Which dimension? Both!??.....

# What's NOT a dimension...

DAMSL dimensions include:

- *'Info-Request'*;
- *'Signal Understanding'* (feedback)

S: Did you say Thursday?

Which dimension? Both!??.....

➔ Info-Request is not a sensible dimension.

3.

DIT++ and ISO 24617-2

# DIT++ taxonomy of communicative functions

Communicative functions:

- A. specific for particular dimensions, e.g. Turn Grab, Stalling, Apology,...:  
“***dimension-specific functions***”
- B. “***general-purpose functions***”, applicable to any dimension of communication, e.g. YN-Question, Inform, Request,...:

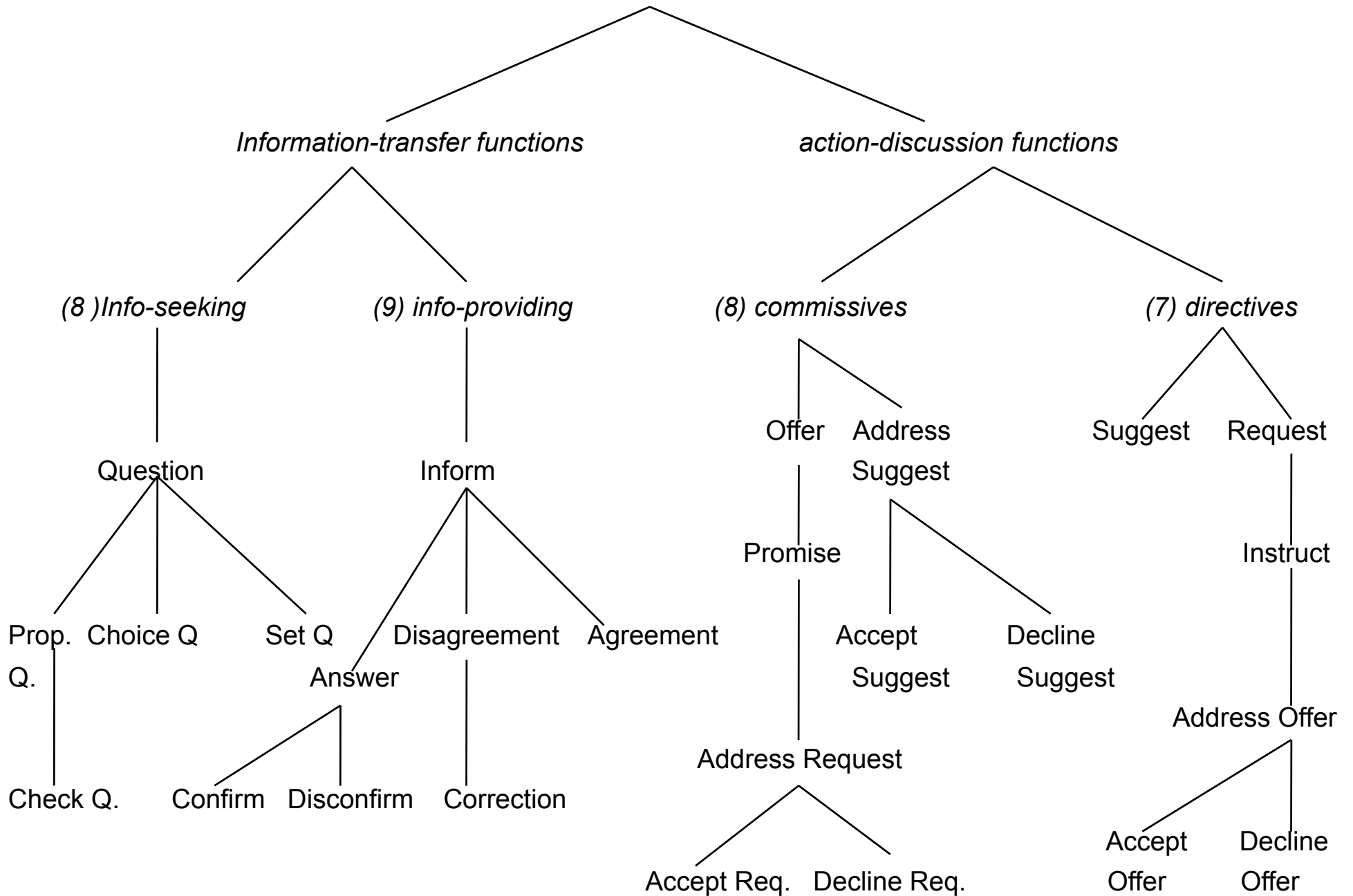
Total of 86 (basic) communicative functions:

- 28 general-purpose functions
- 58 dimension-specific functions

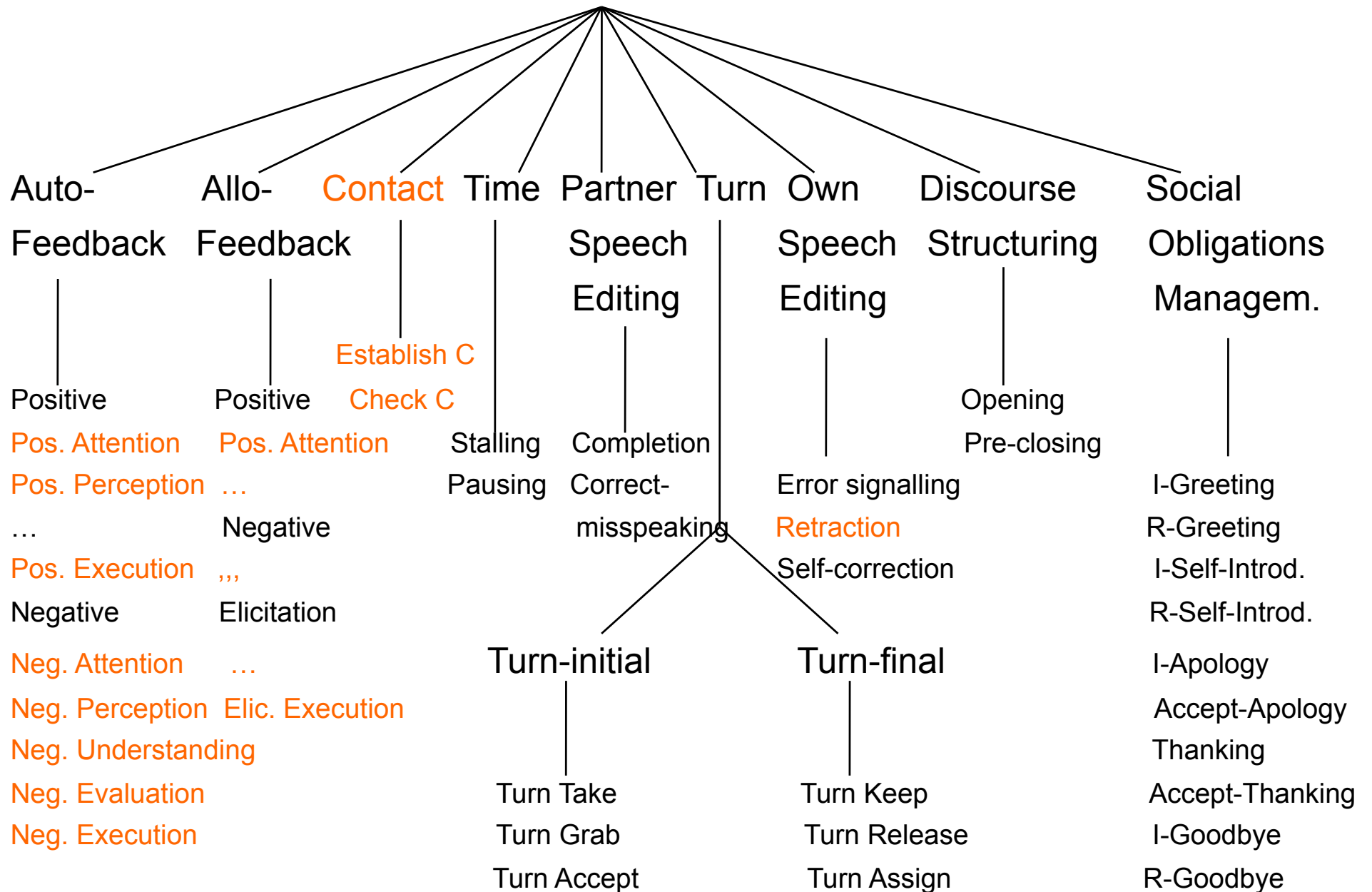
+ ‘function qualifiers’ for variation due to uncertainty, conditionality, sentiment

See <http://dit.uvt.nl>

# general-purpose communicative functions



# Dimension-specific communicative functions in ISO 24617-2 and in DIT++





# Which dimension?

S: Did you say Thursday?

Which dimension? Both!??.....

→ Info-Request is not a sensible dimension.

→ Solution: **Question in the Auto-Feedback dimension**

# Full-blown characterization of dialogue acts

Full-blown characterization of dialogue act involves not only communicative function and semantic content, but also:

- Communicative function **qualifiers**:
  - certainty
  - conditionality
  - sentiment
- **Functional dependence relations**, e.g. *Answer => Question; Decline offer => Offer; Accept apology => Apology;*
- **Feedback relations**: feedback acts at the level of evaluation or execution relate to previous dialogue acts; those at lower levels of processing refer to segments of dialogue;
- **Rhetorical relations**: *Elaborate, Motivate, Exemplify, ...*
  - among dialogue acts
  - among the semantic contents of dialogue acts

# Full characterization of dialogue acts

Some consequences:

A: Where do you live?

B: I live in Toulouse.

A: Do you live in Toulon or in Toulouse?

B: I live in Toulouse.

A: Do you live in Toulouse?

B: I live in Toulouse.

A: You live in Toulouse, don't you?

B: I live in Toulouse.

# Full characterization of dialogue acts

Consequences:

A: Where do you live?

B: I live in Toulouse. *WH-Answer*

A: Do you live in Toulon or in Toulouse?

B: I live in Toulouse. *Choice-Answer*

A: Do you live in Toulouse?

B: I live in Toulouse. *Yes/No-Answer*

DIT++ Release 5: *Answer*, + *functional dependence relation* to the relevant question

# Communicative function **qualifiers**

Variations w.r.t. uncertainty, conditionality, incompleteness, and sentiment, e.g.:

A: Would you like to have some coffee?

B1: Only if you have it ready. (*Conditional Accept Offer*)

B2: That would be wonderful! (*Happily Accept Offer*)

A: Do you know what time the meeting will end?

B: Something like two o'clock maybe. (*Uncertain Answer*)

# Multi-level discourse relations between dialogue units

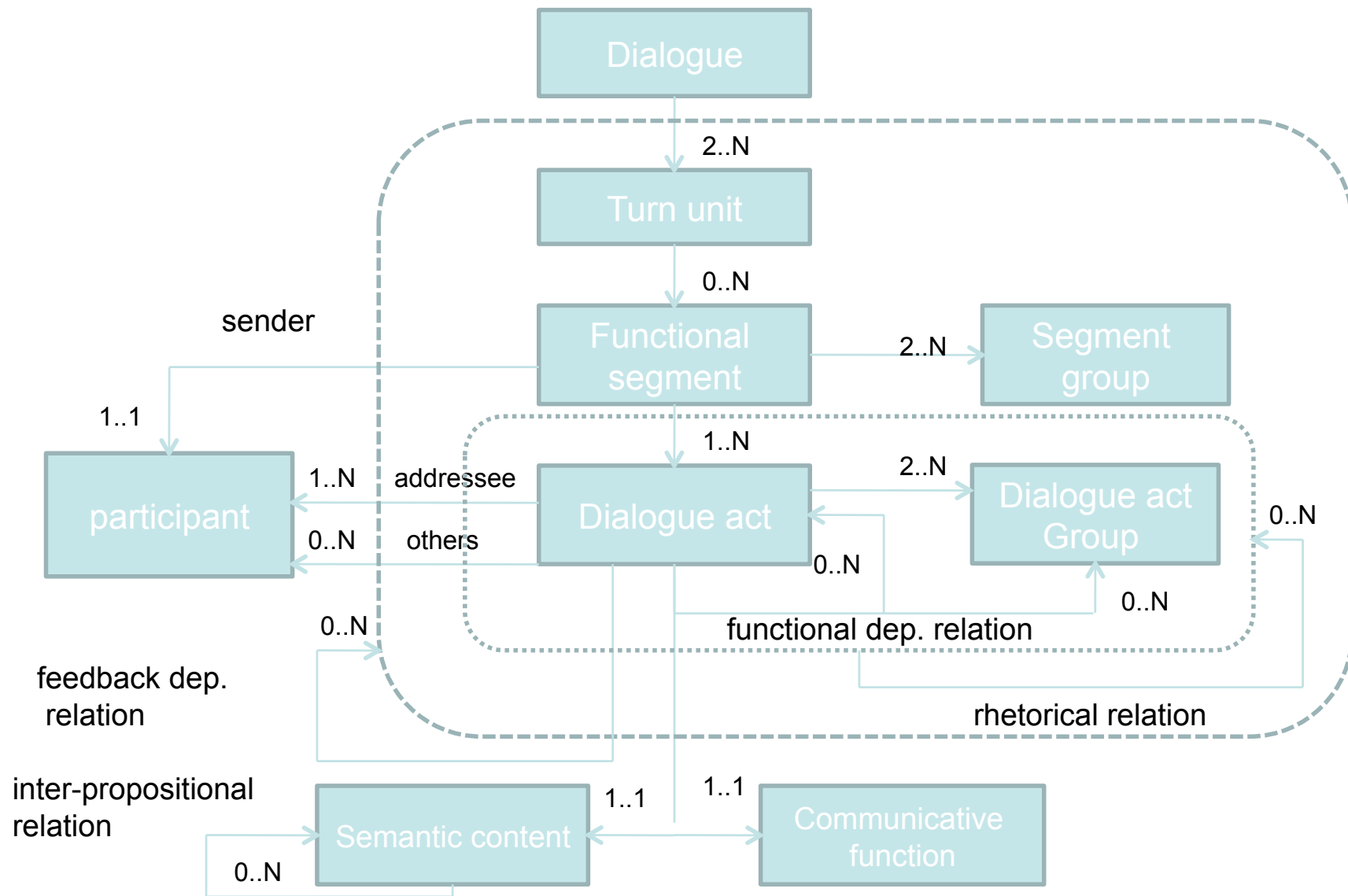
Volha Petukhova\*

Laurent Prévot #

Harry Bunt\*

\* Tilburg Center for Cognition and Communication  
Tilburg University, The Netherlands

# Laboratoire Parole et Langage  
Université de Provence & CNRS



DIT<sup>++</sup> metamodel



# ISO standard 24617-2

- Slightly simplified version of DIT++ taxonomy:
  - 9 dimensions (10);
  - 26 general-purpose communicative functions (28);
  - 30 dimension-specific functions (58)
- Communicative functions defined as ISO data categories, following ISO standard 12620 (see <http://semantic-annotation.uvt.nl/>)
- **Annotation language DiAML** (Dialogue Act Markup Language), with abstract syntax + formal semantics and concrete XML-based syntax
- Accepted as **International Standard, January 11, 2011**
- Expected to be effective (i.e. published) later in 2011





# ISO requirements for annotation standards

*ISO Linguistic Annotation Framework (Ide & Romary, 2005):*

- **annotations:** the linguistic information that is added to segments of language data, independent of the format in which the information is represented;
- **representations:** particular formats in which annotations are rendered, (e.g. in XML, in typed feature structure AVMs, or in graphs in graphical form) independent of their content.

**→ Standards should be formulated at the level of annotations.**

4.

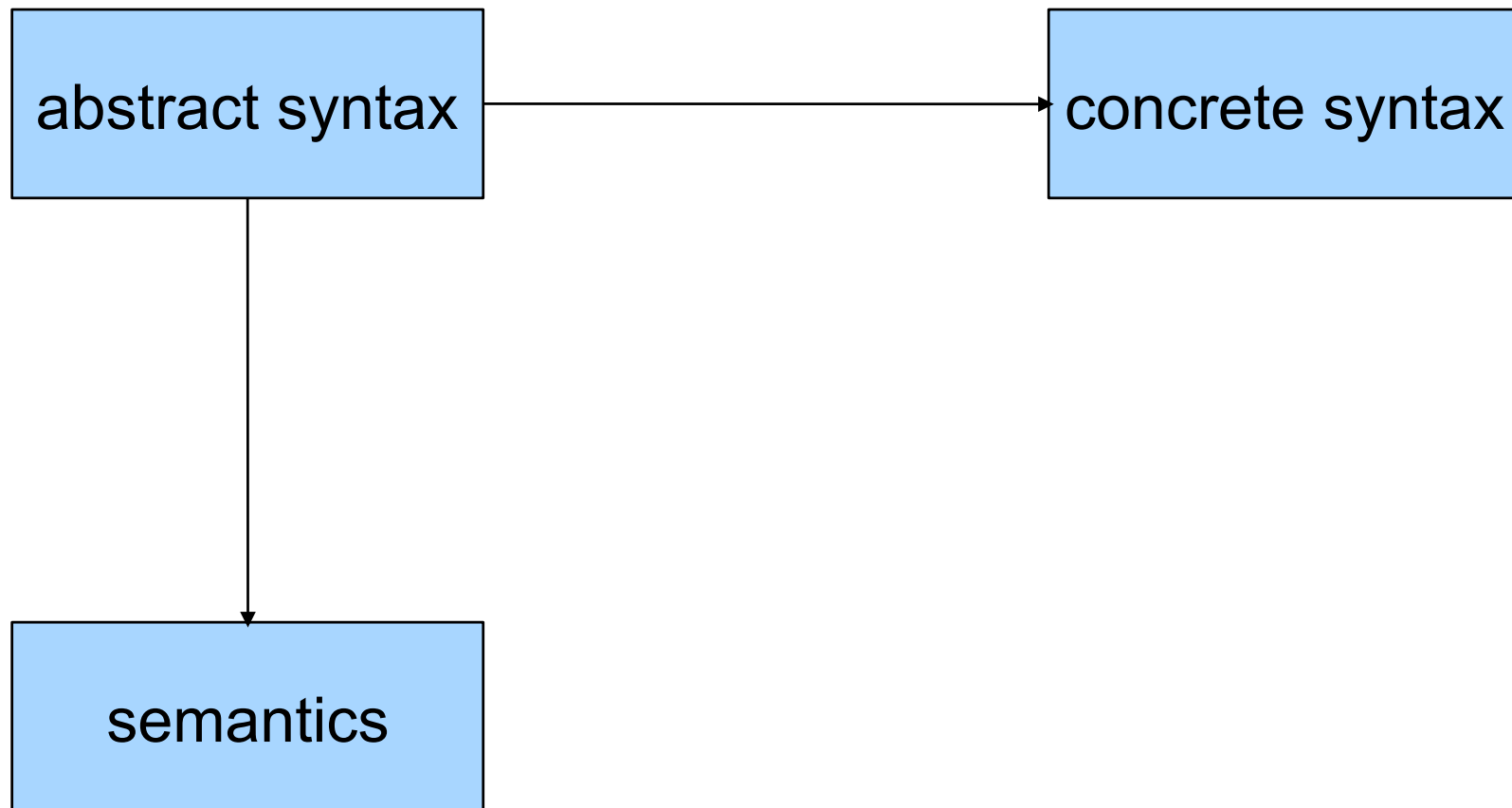
**A Methodology for designing semantic  
annotation schemes**



# Annotation schema design methodology

1. **Conceptual analysis.** Inventory of categories of concepts and relations. “Metamodel”.
2. Specification of **abstract syntax** of annotation language. Formal specification of types of concepts and their combinations (“**annotations**”) in set-theoretical terms.
3. Specification of **semantics** for abstract syntax. Formal specification of how to compositionally compute the meanings of annotation structures.
4. Specification of **concrete syntax**. Definition of concrete, e.g. XML-based, renderings of annotations in some representation format (“**representations**”).

# Annotation/representation distinction: abstract/concrete syntax





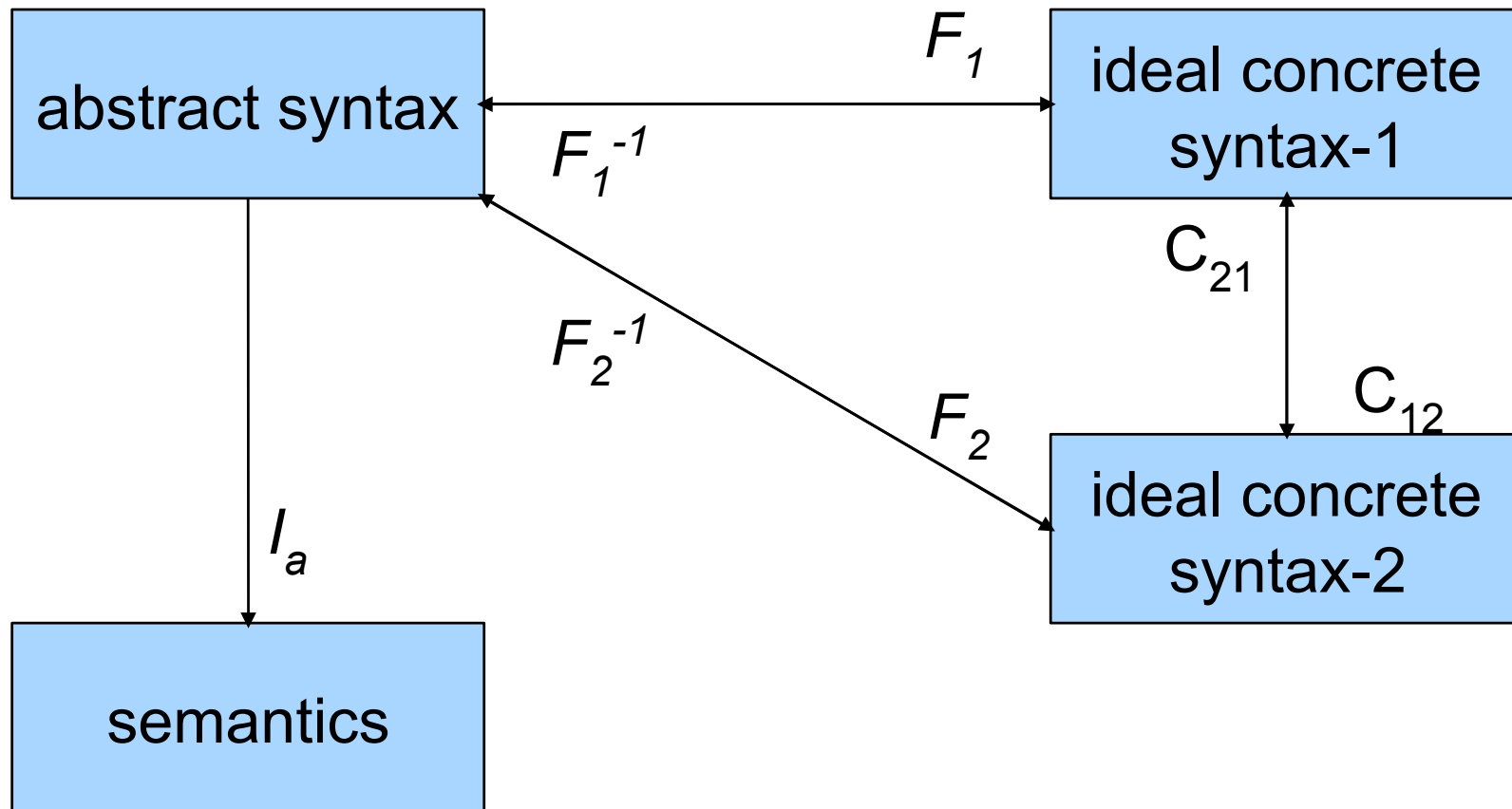
# Ideal Concrete Syntax

An *ideal concrete syntax* defines an ***ideal representation format***:

1. For every annotation structure, defined by the abstract syntax, the concrete syntax defines a representation;
2. Every representation, defined by the concrete syntax, is the rendering of a unique annotation structure according to the abstract syntax.



# Ideal concrete syntax



## 5. DiAML

Dialogue Act Markup Language  
abstract and concrete syntax;  
semantics



# DiAML abstract syntax

An annotation structure is a pair  $\langle E, L \rangle$  consisting of a set  $E$  of **entity structures** and set  $L$  of **link structures**, which connect entity structures.

- *Entity structures* contain semantic information about a segment of primary data.
- *Link structures* describe semantic relations between segments of primary data.

Most important type of entity structure in DiAML:

**dialogue act structure**  $\langle S, A, d, f \rangle$ , consisting of:

- speaker  $S$ ;
- addressee  $A$ ;
- dimension  $D$ ;
- communicative function  $f$  or a pair  $\langle f, q \rangle$  with qualifier(s)  $q$

Link structures represent functional, feedback, and rhetorical relations.



# DiAML example, *segmentation*

1. P1: ***What time does the next train to Tilburg leave?***

Task: **fs1**: *What time does the next train to Tilburg leave?*

**Set-Question ('WH-question')**

2. P2: ***The next train to Tilburg leaves I think at 8:32.***

Task: **fs2**: *The next train to Tilburg leaves I think at 8:32.*

**Answer [uncertain]**

AuFB: **fs3**: *The next train to Tilburg leaves*

**Positive AutoFeedback**



# DiAML concrete syntax, example

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
  sender="#p1" addressee="#p2"
  communicativeFunction="setQuestion" dimension="task" />
<dialogueAct xml:id="da2" target="#fs2.1"
  sender="#p2" addressee="#p1"
  communicativeFunction="answer" dimension="task"
  certainty="uncertain"/>
<functionalDependence dact="#da2" functAntecedent="#da1"/>
<dialogueAct xml:id="da3" target="#fs2.2"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact="#da3" fbSegment="#fs1"/>
</diaml>
```

6.

**DiAML semantics**

# DiAML semantics

Remember: **dialogue act structure**  $\langle S, A, d, f \rangle$

The interpretation of a dialogue act structure is a function that can be applied to a semantic content, resulting in the specification of a context update operation.

Implementation: recursive **valuation function**  $V$ , recursion ending at application of model assignment  $F$ :

$$V(\langle S, A, d, f \rangle) = V(f)(\langle F(S), F(A), F(d) \rangle)$$

Taxonomy structure of communicative functions  $\rightarrow$  Specification of  $V(f)$  as combinations of *elementary update schemes*

Elementary update schemes: e.g.

$U_1$ : *Given certain information  $p$ , add to the addressee's pending context that the speaker believes that  $p$*

# DiAML semantics, example

User: Will KLM flight 476 arrive at seven fifteen?

Propositional Question (“Yes/No question”), semantics:

$V(\text{PropositionalQuestion}) = \lambda X, Y, D, p. [U_{10}(X, Y, D^*_Y, p) \cup U_{11}(X, Y, D^*_Y, p)]$

$U_{10}(X, Y, D^*, z) =$  add to  $D_{Y^*}$  that  $X$  wants to know whether  $z$

$U_{11}(X, Y, D^*, z) =$  add to  $D_{Y^*}$  that  $X$  assumes that  $Y$  knows whether  $z$

$V(\langle S, A, d, f \rangle) = V(f)(\langle S', A', d' \rangle) =$

$= \lambda X, Y, D, p. [U_{10}(X, Y, D^*_Y, p) \cup U_{11}(X, Y, D^*_Y, p)]$  applied to  
 $Usr, Sys, Task, ArrTime(KL476, 19:15)$

Result: (1) Sys:  $Usr$  wants to know whether  $ArrTime(KL476, 19:15)$ ;

(2) Sys:  $Usr$  assumes Sys knows whether  $ArrTime(KL476, 19:15)$

# Context models/information states

**Content** of a context model / information state:

*All and exactly that information which has to be updated when interpreting a dialogue act.*

Formalization and implementation of a context model:

- DRSs (Poesio & Traum, 1998)
- ‘Contexts’ in Constructive Type Theory (Ahn, 2001)
- ‘Modular Partial Models’ (Bunt, 2002)
- ‘Dependent Record Types’ (Cooper, 2004)
- Typed Feature Structures (Keizer & Bunt, 2007)

Common assumption: context models are highly structured, to facilitate efficient updating.

**Pending context**”: information “under discussion”, which has yet to be evaluated for consistency with “permanent” context.

Context update = addition of elements to pending context.

# DIT context models

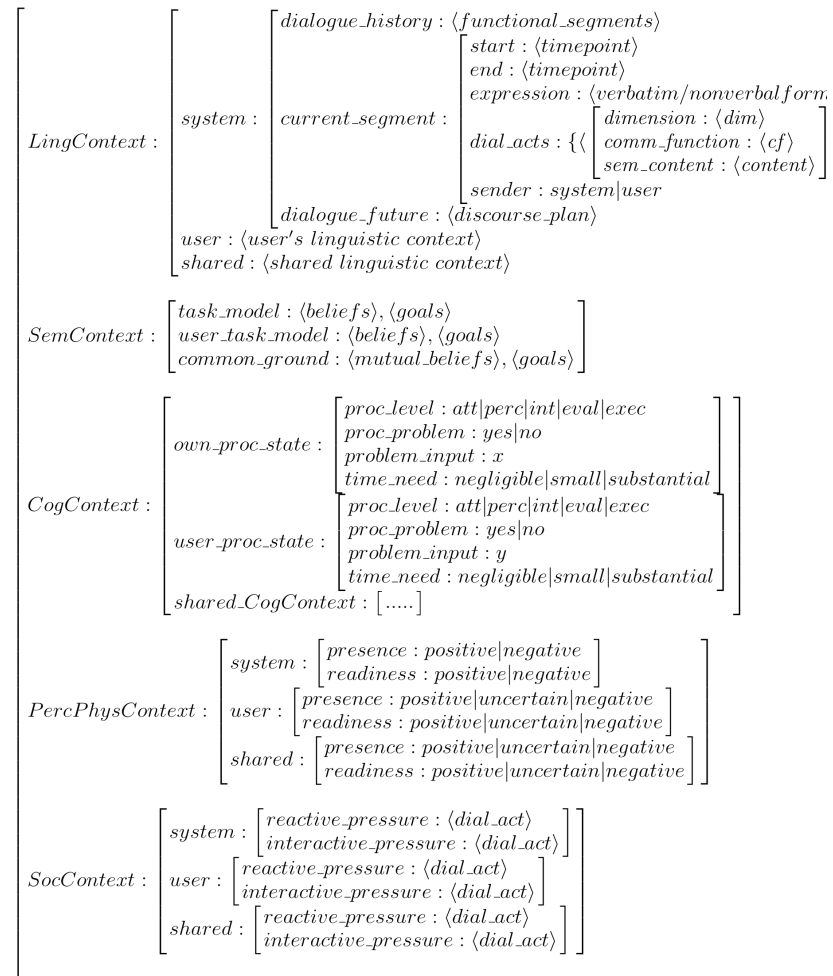
Content of context models containing the kinds of information required by the 10 dimensions of DIT is best structured into 5 components to facilitate efficient representation and updating:

- **Linguistic** context: dialogue history; dialogue future
- **Cognitive** context: agent's own processing of previous utterances; beliefs about partner's processing
- **Semantic** context: information about the task (domain)
- **Perceptual/physical** context: perceptual information (e.g. visual)
- **Social** context: communicative obligations and permissions

**“Pending context”**: information “under consideration”, which has yet to be evaluated for being consistent with content of “permanent” context.

Context update = addition of elements to pending context.

# DIT context model implementation





# Example: feedback act with general-purpose function

B: (u1)

A: Could you please repeat that?

$F(\text{Request}) =$

$$\lambda C. \lambda X. \lambda Y. \lambda D. \lambda \alpha. [U_{23}(X, Y, D, \alpha, C) \cup U_{26}(X, Y, D, \alpha, C)]$$

Unconditional request:  $C = T$  (the universally true statement)

$F(\text{Request})(A, B, \text{Autofeedback}, \text{Repeat}(u1), T) =$

$= U_{23}(A, B, \text{CogContext}, \text{Repeat}(u1), T) \cup U_{26}(A, B, \text{CogContext}, \text{Repeat}(u1), T) =$

$B^*_{\text{CogC}} =+ \mathbf{Bel}(B, \mathbf{Want}(A, [\mathbf{WillDo}(B, \text{Repeat}(u1)) \rightarrow \mathbf{CommitDo}(B, \text{Repeat}(u1))]));$

$B^*_{\text{CogC}} =+ \mathbf{Bel}(B, \mathbf{Bel}(A, \mathbf{CanDo}(B, \text{Repeat}(u1))))$

# Example: Turn management act with dimension-specific function

A: Charlie?

$$\begin{aligned} F(\text{TurnAssign})(A, B) &= [\lambda X. \lambda Y. U_{101}(X, Y, \text{TurnM}) \cup U_{102}(X, Y, \text{TurnM})](A, B) = \\ &= U_{101}(A, B, \text{TurnM}) \cup U_{102}(A, B, \text{TurnM}) = \\ &= B^*_{\text{LingC}} =+ \mathbf{Bel}(B, \mathbf{Current-Speaker}(A)) \\ &B^*_{\text{LingC}} =+ \mathbf{Bel}(B, \mathbf{Want}(A, \mathbf{Next-Speaker}(B))) \end{aligned}$$

i.e. B believes that A currently has the speaker role;  
B believes that A wants B to next have the speaker role

7.

Conclusion

# Concluding remarks; Perspectives

- ISO 24617-2 accepted in January 2011 as international standard.
- DIT<sup>++</sup> Release 5 is a strictly downward compatible extension of ISO 24617-2
- Distinction of general-purpose versus dimension-specific function and 9 or 10 dimensions, in combination with qualifiers, functional dependence relations, feedback relations, and rhetorical relations gives ISO 24617-2 and DIT<sup>++</sup> the means to build annotations with a high precision; the higher levels in the communicative function hierarchies allow less fine-grained annotations.
- The DiAML annotation language has a formal model-theoretic semantics semantics in terms of updates of a context model, using combinations of elementary update schemes.
- ISO 24617-2/DIT<sup>++</sup> annotations can be made manually with satisfactory reliability, and can be automatically assigned with high accuracy (Petukhova & Bunt, IWCS 2011; Petukhova PhD, August 2011).



7.

## References



## References

- Bunt, Harry (2000) Dialogue pragmatics and context specification. In Harry Bunt and William Black (eds.), *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*, pp. 81-150. Amsterdam: John Benjamins.
- Bunt, Harry (2009) Multifunctionality and multidimensional dialogue semantics. In *Proceedings of DiaHolmia, 13th Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm, pp. 3-14.
- Bunt, Harry (2010) A methodology for designing semantic annotation languages. In *Proceedings of the 2nd International Conference on Global Interoperability for Language Resources (ICGL-2)*, Hong Kong, January 2010, pp. 29-46.
- Bunt, Harry (2011) Multifunctionality in dialogue. *Computer, Speech and Language* 25, 225-245.
- Bunt, Harry (2011) The semantics of dialogue acts. In *Proceedings IWCS 2011, the 9th International Conference on Computational Semantics*. Oxford.
- Bunt, Harry, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum (2010). Towards an ISO standard for dialogue act annotation. In *Proceedings 7th International Conference on Language Resources and Evaluation (LREC 2010)*. Paris: ELRA.
- Ide, Nancy and Harry Bunt (2010) Anatomy of semantic annotation schemes: Mappings to GrAF. In *Proceedings of the 4th Linguistic Annotation Workshop (LAW-IV)*, Uppsala.
- Petukhova, Volha and Laurent Prévot and Harry Bunt (2011) Multilevel discourse relations between dialogue units. In *Proceedings ISA-6, the 6th International Workshop on Interoperable Semantic Annotation*. Oxford.
- Poesio, Massimo and David Traum (1998) Towards an axiomatisation of dialogue acts. In *Proceedings of the Twente Workshop on the Semantics and Pragmatics of Dialogue*, Enschede, pp. 207-222.

# Web sites

<http://semantic-annotation.uvt.nl>

information about recent ISO projects  
and other efforts to develop semantic  
annotation standards

<http://dit.uvt.nl>

DIT<sup>++</sup> annotation scheme; definitions  
of communicative functions;  
examples; bibliography

■