

Annotating Discontinuous Structures in XML: the Multiword Case

Emanuele Pianta and Luisa Bentivogli

ITC-irst, Trento, Italy

- XML and overlapping trees
- Annotating lexical information
- Annotating (discontinuous) multiwords
- 4 annotation schemes
- Conclusions

- PRO: trees are a very common formalism for various linguistic representation levels:
 - syntax
 - text divisions (chapters, sections, paragraphs)
 - structure of the content (e.g. RST)
 - graphical layout ...
- CONS:
 - one XML document <-> one tree structure
 - no natural representation of multiple overlapping hierarchies

- *Multiple* linguistic representations
 - Ex: c-structure, f-structure, s-structure (LFG)
- *Logical vs. physical* (layout) structure of the text
 - Ex: a sentence spanning over one line and a half (poetry)
- *Alternative* representations for the same level
 - Ex: alternative parse trees
 - Ex: constituent-based vs dependency-based syntax

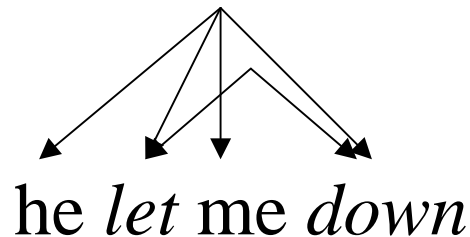
One tree, discontinuous structures



- Discontinuous units
 - Phrasal and separable verbs
 - Ex: don't *let me down*
 - Ex: Ich *fahrte am Morgen ab*
 - Non-contiguous multiword expressions
 - Ex: people should *take it really easy*
- Long distance dependences
 - Pronoun antecedents
 - Ex: I saw *Mary* yesterday. *She* was laughing.

Why are Discontinuous Structures a problem for XML annotation?

- Apparently only *one* representation level
- But in all non trivial annotation tasks *more than one* linguistic level is involved
- Minimal level: text as an ordered sequence of graphical words
- Discontinuous word units implies overlapping tree branches



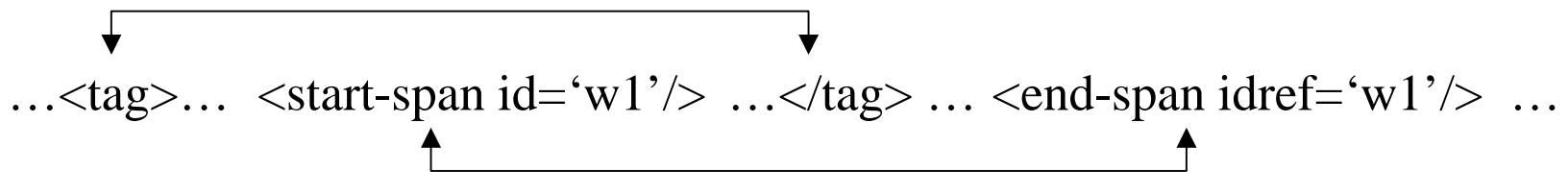
- SGML solution: the CONCUR feature
 - Allows for specifying multiple DTDs
 - Processing one document per time
 - Not available in XML
 - Optional, not supported by all SGML processors

- *Multiple* encoding
 - Straightforward
 - Redundant
 - Risk of loosing alignment between encoding
- *Milestones*: marking an alternative (ghost) tree with empty elements

↓ ↓

...<tag>... <start-span id='w1'/> ...</tag> ... <end-span idref='w1'/> ...

↑ ↑

The diagram shows a sequence of XML elements: "...<tag>... <start-span id='w1'/> ...</tag> ... <end-span idref='w1'/> ...". Above the sequence, a horizontal line with two downward-pointing arrows connects the first and second elements. Below the sequence, a horizontal line with two upward-pointing arrows connects the second and fourth elements.

Cons: ad-hoc processing, representations with different status

- *Stand-off* annotation:
 - keeping annotation separated from the text
 - in the same file or in different file (cfr GATE)
 - pointers

PROS:

- elegance and clarity
- processing conceptually simple

CONS:

- physical separation between text and annotation
- complexity of pointer processing

The multiword case study



Multiword: a lexical unit composed of more than one word (idioms and restricted collocations)

andarci piano (lit. *go-there slowly*)

take it easy

Coi superalcolici bisogna *andarci* veramente *piano*

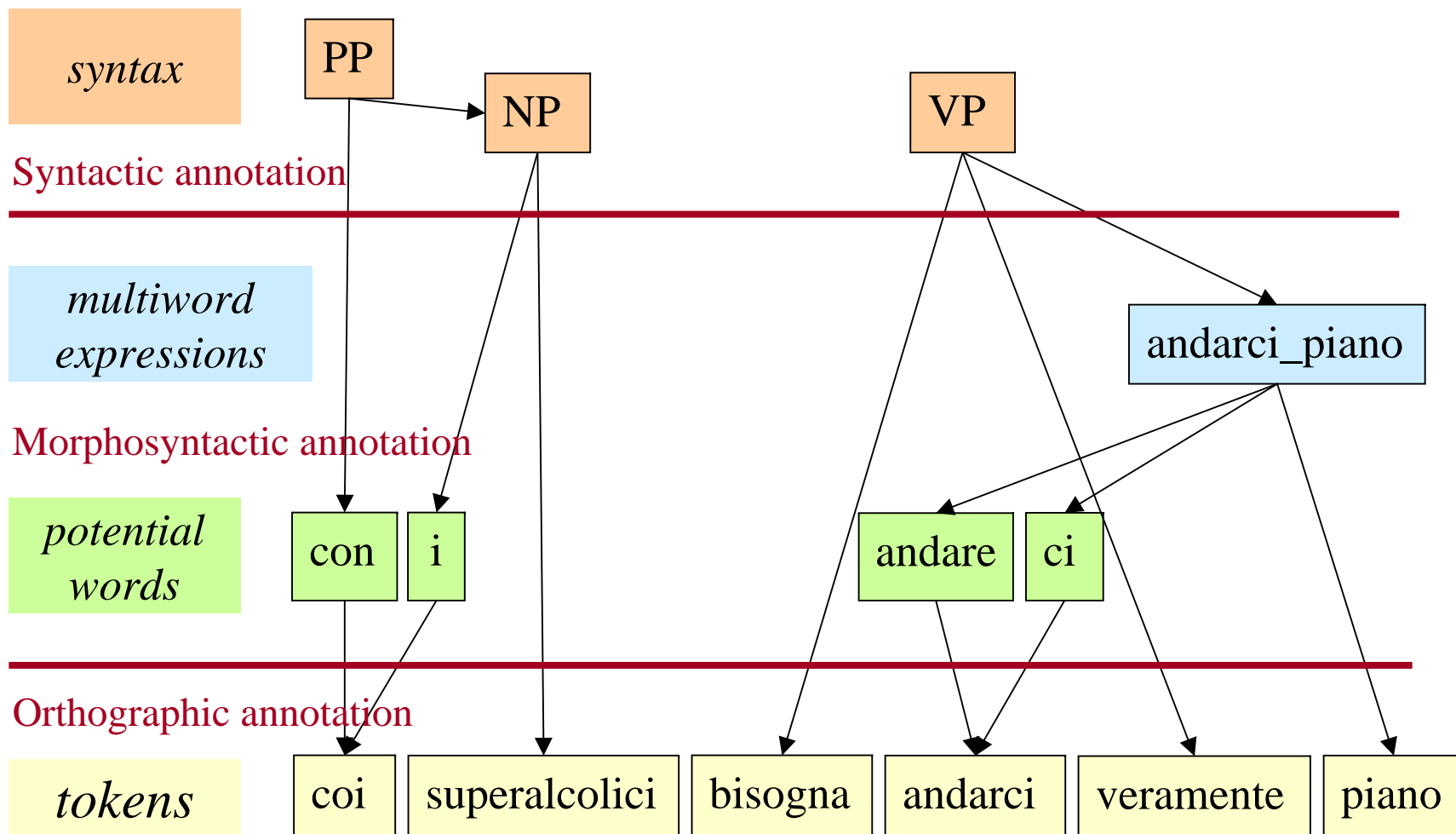
People should *take it* really *easy* with liquors

3 Lexical representation levels



- *Tokens*: graphical words
EX: andarci veramente piano
- *Potential Words*: inflected word forms before phonological and/or graphical adjustment
EX: andare ci veramente piano
- *Lexical Units*: one or more potential words carrying a unitary meaning
EX: andarci_piano veramente
- Conceptually distinct (levels of linguistic annotation)
- Interacting in complex way

Interaction between lexical representation levels



- A multi-level linguistically annotated corpus
 - Structure of the text, orthographic, morphosyntactic, multiwords, named entities, syntactic, word senses
- XML format following ISO/TC 37/SC 4 standard for linguistic resources (Ide and Romary, 2002)
 - Structures: nestable <struct> elements
 - attributes: <feat> elements
 - Stand-off annotation
 - XLink, XPointer, IDREFs

Orthographic Annotation

Coi superalcolici bisogna **andarci** veramente **piano**
*People should **take it** really **easy** with liquors*

```
<struct type="ortho">
```

```
...
```

```
<struct type="t-level" id="t_4">
```

```
<feat type="token">andarci</feat>
```

```
<feat type="case">lower</feat>
```

```
<seg startsAt="62" endsAt="68"/>
```

```
</struct>
```

```
<struct type="t-level" id="t_5">
```

```
<feat type="token">veramente</feat>
```

```
<feat type="case">lower</feat>
```

```
<seg startsAt="69" endsAt="77"> </seg>
```

```
</struct>
```

```
...
```

```
</struct>
```

Morphosyntactic Annotation

```
<struct type="morpho" xml:base="../ortho/ministero-ort.xml">
```

...

```
<struct type="w-level" id="w_5" xlink:href="#xpointer(id('t_4'))">
```

```
<feat type="lemma">andare</feat>
```

```
<feat type="stem">and</feat>
```

```
<feat type="form">andar</feat>
```

```
<feat type="pos">v</feat>
```

```
<feat type="mood">inf</feat>
```

```
<feat type="tense">pres</feat>
```

```
</struct>
```

```
<struct type="w-level" id="w_6" xlink:href="#xpointer(id('t_4'))">
```

```
<feat type="lemma">ci</feat>
```

```
<feat type="pos">pron</feat>
```

```
</struct>
```

...

```
</struct>
```

“...andarci...”

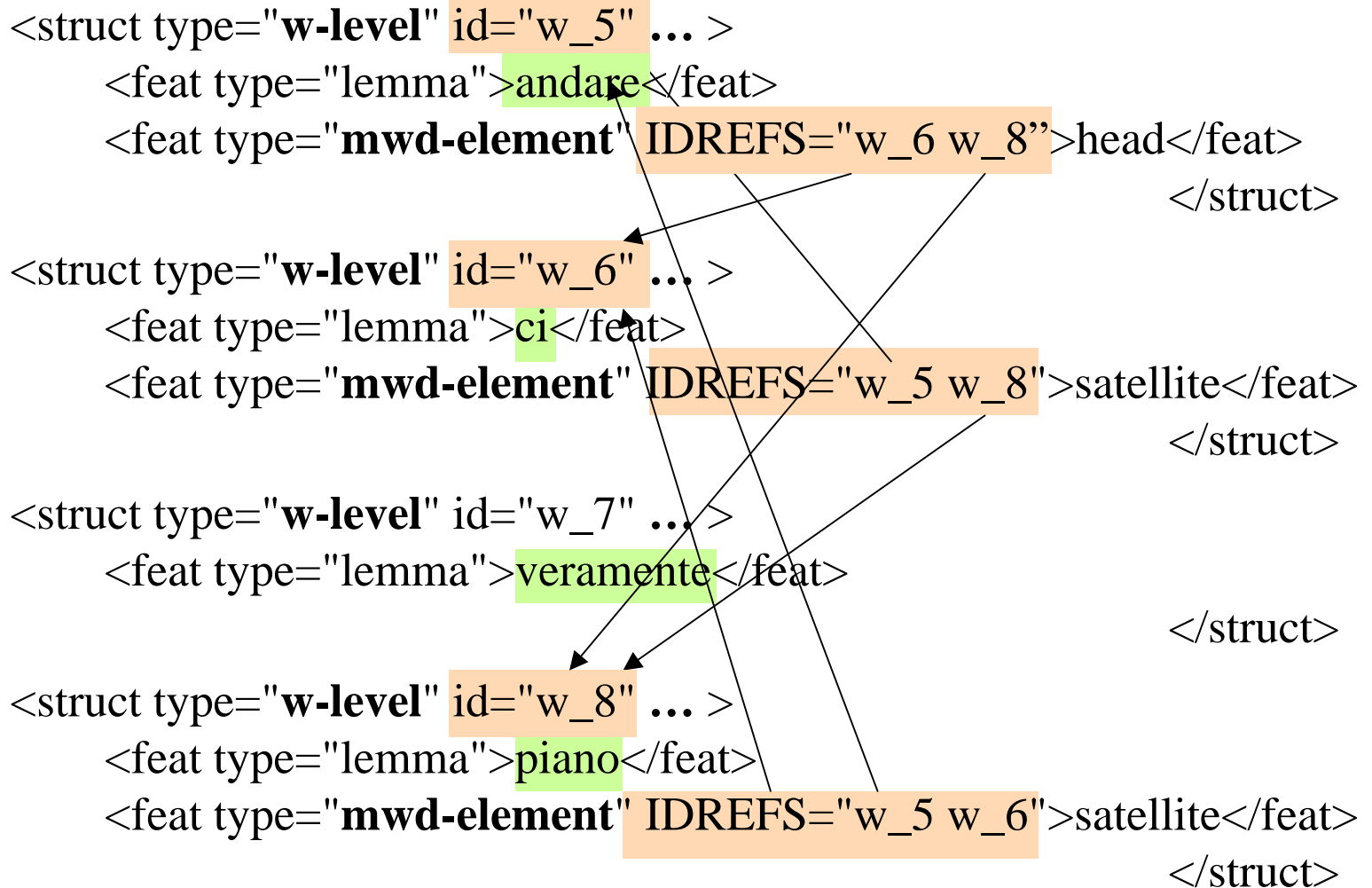
“...*take it*...”

Continuous Multiwords Scheme A (in-line, hierarchical)



```
<struct type="mwd-level" id="mwd_1">
  <feat type="lemma">andarci_piano</feat>
  <feat type="pos">v</feat>
  <struct type="w-level" id="w_5" xlink:href="#xpointer(id('t_4'))">
    <feat type="lemma">andare</feat>
    <feat type="pos">v</feat>
    <feat type="mwd-function">head</feat></struct>
  <struct type="w-level" id="w_6" xlink:href="#xpointer(id('t_4'))">
    <feat type="lemma">ci</feat>
    <feat type="pos">clitic</feat>
    <feat type="mwd-function">satellite</feat></struct>
  <struct type="w-level" id="w_7" xlink:href="#xpointer(id('t_5'))">
    <feat type="lemma">piano</feat>
    <feat type="pos">adv</feat>
    <feat type="mwd-function">satellite</feat></struct>
</struct>
```

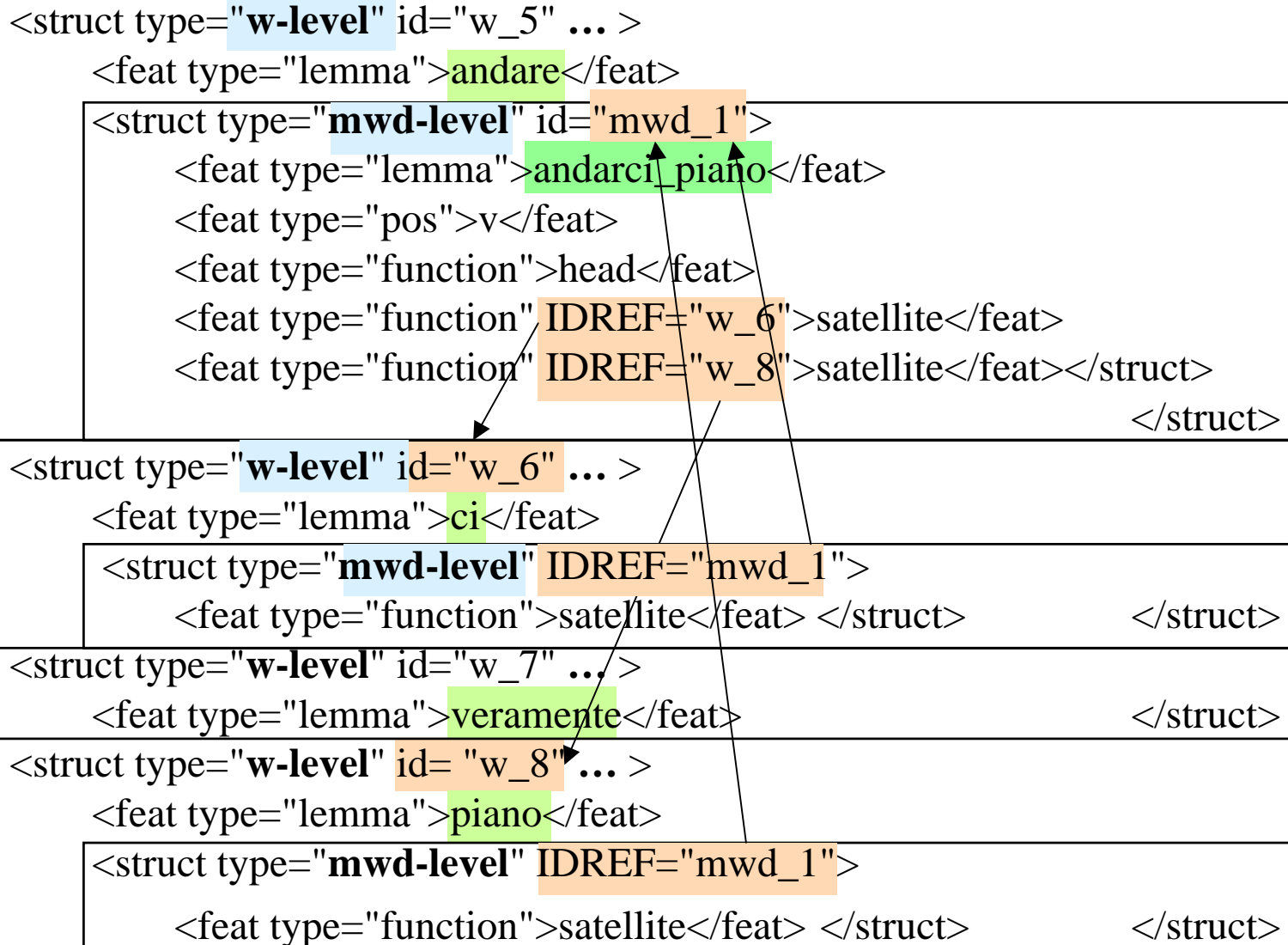
Discontinuous Multiwords Scheme B (in-line, flat)



- No explicit multiword-level
 - Multiword information is represented at word level:
 - Through pointers inter-connecting the various parts of each multiword
- ```
<feat type="mwd-element" IDREFS="w_6 w_8">
```
- Pros:
    - Structural simplicity
  - Cons:
    - Proliferation of pointers
    - Lack of a specific structure to represent information that pertains to the multiword as a unit, e.g. lemma and PoS

# Discontinuous Multiwords Scheme C (in-line, inverse)

```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <feat type="function">head</feat>
 <feat type="function" IDREF="w_6">satellite</feat>
 <feat type="function" IDREF="w_8">satellite</feat></struct>
 </struct>
<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct>
 </struct>
<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat>
 </struct>
<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct>
 </struct>
```



- The multiword-level structure is *included* in the word-level structure
  - Pointers to the components of the multiword
- PROS:
  - Explicit multiword level
  - in-line annotation
- Cons:
  - Nesting conceptually complex structures within simple ones may be incorrect/inelegant

# Discontinuous Multiwords Scheme D (stand-off) - I



*potential word  
section*

```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <feat type="mwd-element">head</feat>
```

</struct>

```
<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <feat type="mwd-element">satellite</feat>
```

</struct>

```
<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat>
```

</struct>

```
<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <feat type="mwd-element">satellite</feat>
```

</struct>

# Discontinuous Multiwords Scheme D (stand-off) - II

*multiword  
section*

```
<struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <struct type="mwd-element" IDREF="w_5">
 <feat type="function">head</feat>
 </struct>
 <struct type="mwd-element" IDREF="w_6">
 <feat type="function">satellite</feat>
 </struct>
 <struct type="mwd-element" IDREF="w_8">
 <feat type="function">satellite</feat>
 </struct>
</struct>
```

# Discontinuous Multiwords

## Scheme D (stand-off) – I & II



```
<struct type="w-level" id="w_5" ...>
 <feat type="lemma">andare</feat>
 <feat type="mwd-element">head</feat></struct>
<struct type="w-level" id="w_6" ...>
 <feat type="lemma">ci</feat>
 <feat type="mwd-element">satellite</feat></struct>
<struct type="w-level" id="w_7" ...>
 <feat type="lemma">veramente</feat></struct>
<struct type="w-level" id="w_8" ...>
 <feat type="lemma">piano</feat>
 <feat type="mwd-element">satellite</feat></struct>
```

*potential word  
section*

---

```
<struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <struct type="mwd-element" IDREF="w_5">
 <feat type="function">head</feat></struct>
 <struct type="mwd-element" IDREF="w_6">
 <feat type="function">satellite</feat></struct>
 <struct type="mwd-element" IDREF="w_8">
 <feat type="function">satellite</feat></struct>
</struct>
```

*multiword  
section*

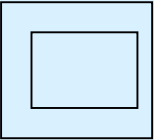
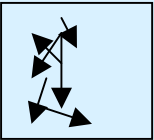
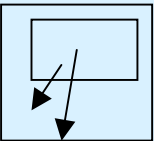
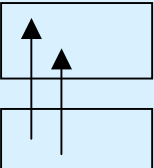
# Scheme D (stand-off): Pros and Cons



- Word-level and multiword-level in two different sections
- The status of a word as element of a multiword is marked explicitly in the potential word section
- Information pertaining to the multiword level can be retrieved starting from the first section, by following the ID-IDREF link backward with an XPATH expression
- stand-off syntactic annotation can point to unitary multiword level structures in the multiword section of the annotation
- If we have two different files instead of the same file, annotation scheme D can still be applied substituting the IDREFs with XLinks and XPointers.

# 4 Annotation Schemes: summary



	in/off	discont. mwd	explicit mwd lev.	pointer complex.	clarity / elegance
A 	in-line	no	yes	none	☺
B 	in-line	yes	no	many	☹
C 	in-line	yes	yes	some	☹
D 	stand-off	yes	yes	few	☺

- Representing *multiple/overlapping trees* challenges the expressive power of XML
- Annotating *discontinuous units* is part of the same problem
- Three ling. levels are involved in lexical annotation:
  - *tokens, potential words, lexical units*
- Four alternatives for representing multiwords
  - stand-off annotation offers the best solution for representing *discontinuous multiwords*



# ORIGINALE!!!

## (In-line annotation - Scheme B)



```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <feat type="mwd-element" IDREFS="w_6 w_8">head</feat>
</struct>

<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <feat type="mwd-element" IDREFS="w_5 w_8">satellite</feat>
</struct>

<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat>
</struct>

<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <feat type="mwd-element" IDREFS="w_5 w_6">satellite</feat>
</struct>
```

# ORIGINALE!!!

## (In-line annotation - Scheme C)



```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <feat type="function">head</feat>
 <feat type="function" IDREF="w_6">satellite</feat>
 <feat type="function" IDREF="w_8">satellite</feat></struct></struct>
<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct></struct>
<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat></struct>
<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct></struct>
```

# Discontinuous Multiwords (In-line annotation - Scheme C)

```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <feat type="function">head</feat>
 <feat type="function" IDREF="w_6">satellite</feat>
 <feat type="function" IDREF="w_8">satellite</feat></struct> </struct>
 <struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 </struct>
 <struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat></struct>
 <struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <feat type="function">satellite</feat> </struct> </struct>
```

# Discontinuous Multiwords (In-line annotation - Scheme C)

```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <struct type="mwd-level" id="mwd_1">
 <feat type="lemma">andarci_piano</feat>
 <feat type="pos">v</feat>
 <feat type="function">head</feat>
 <feat type="function" IDREF="w_6">satellite</feat>
 <feat type="function" IDREF="w_8">satellite</feat></struct> </struct>
<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct> </struct>
<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat></struct>
<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <struct type="mwd-level" IDREF="mwd_1">
 <feat type="function">satellite</feat> </struct> </struct>
```

# Discontinuous Multiwords -2 (Scheme D)

```
<struct type="w-level" id="w_5" ... >
 <feat type="lemma">andare</feat>
 <feat type="mwd-element">head</feat>
</struct>
```

```
<struct type="w-level" id="w_6" ... >
 <feat type="lemma">ci</feat>
 <feat type="mwd-element">satellite</feat>
</struct>
```

```
<struct type="w-level" id="w_7" ... >
 <feat type="lemma">veramente</feat>
</struct>
```

```
<struct type="w-level" id="w_8" ... >
 <feat type="lemma">piano</feat>
 <feat type="mwd-element">satellite</feat>
</struct>
```