

TERENCE WP4

for reasoning about annotations and
generating textual smart games

Rosella Gennari

KRDB - CS Faculty - UniBZ

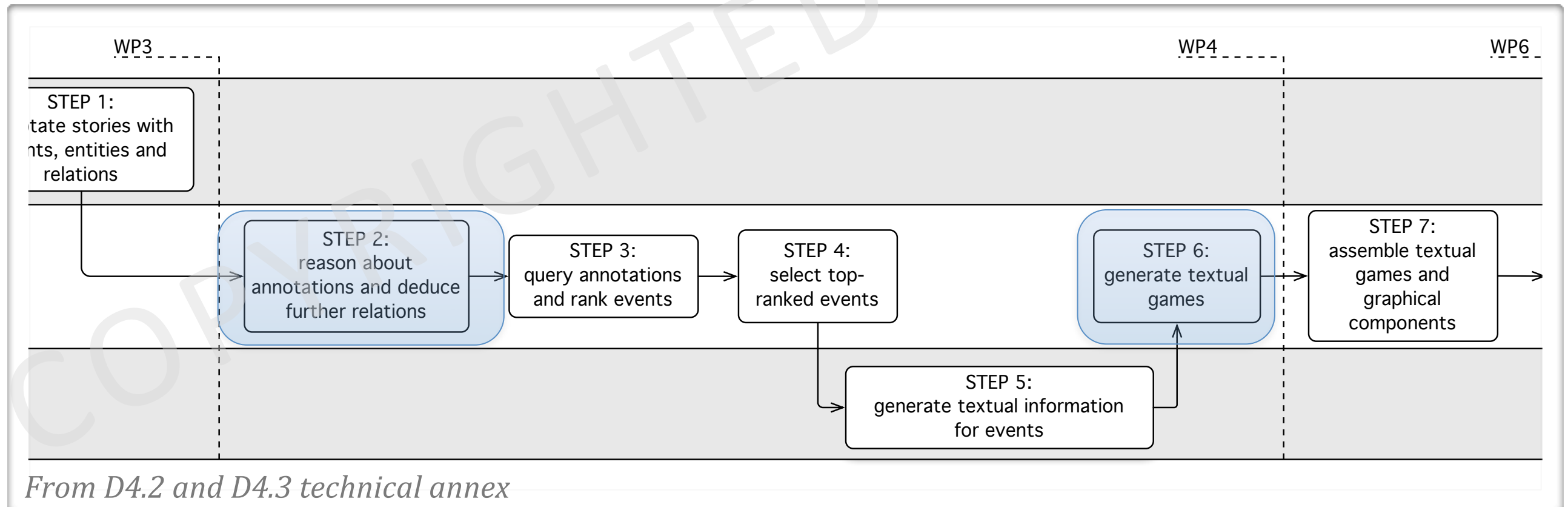
<http://www.terenceproject.eu>



TERENCE

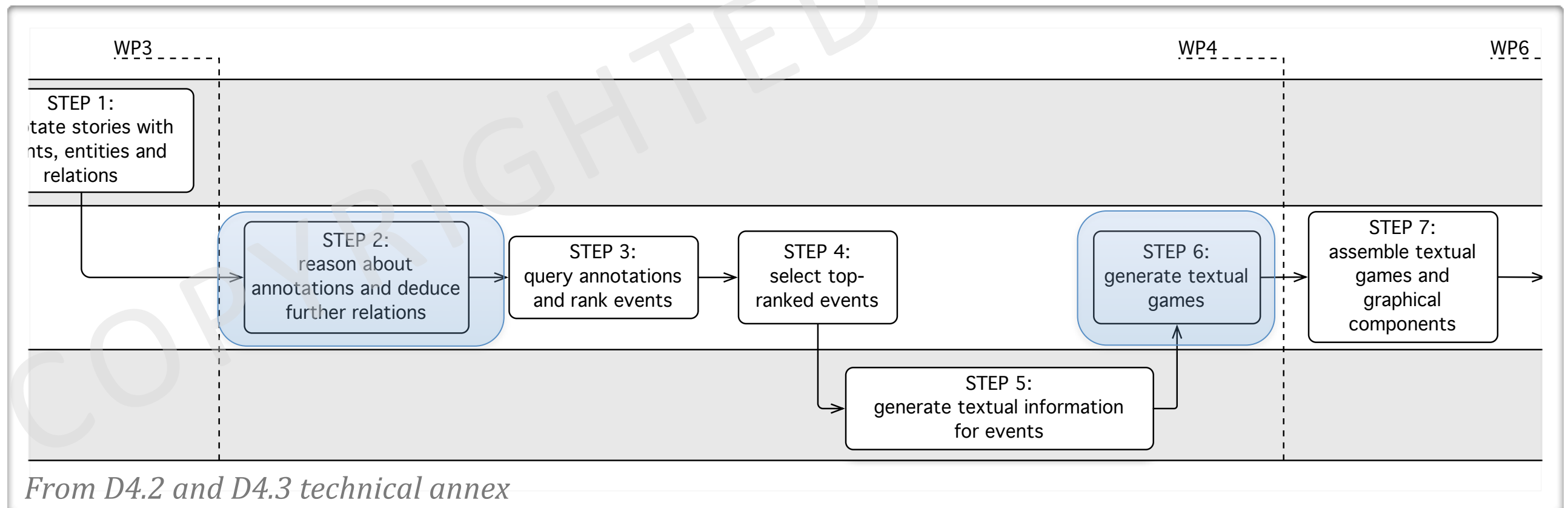
A series of colorful, wavy lines in shades of blue, yellow, and red, flowing across the bottom of the slide.

and its deliverables



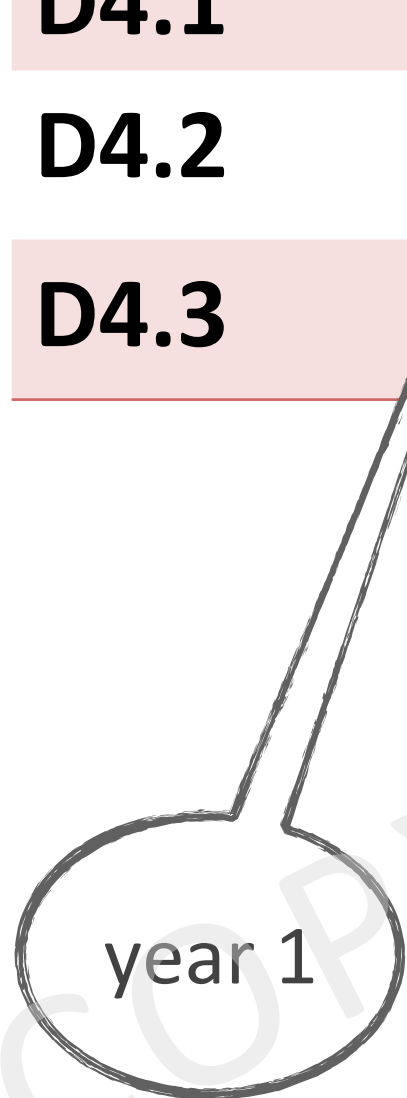
and its deliverables

Deliverable	1 st release	2 nd release	3 rd release
D4.1	Sep. 2011		
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013



and its deliverables

Deliverable	1 st release	2 nd release	3 rd release
D4.1	Sep. 2011		
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013



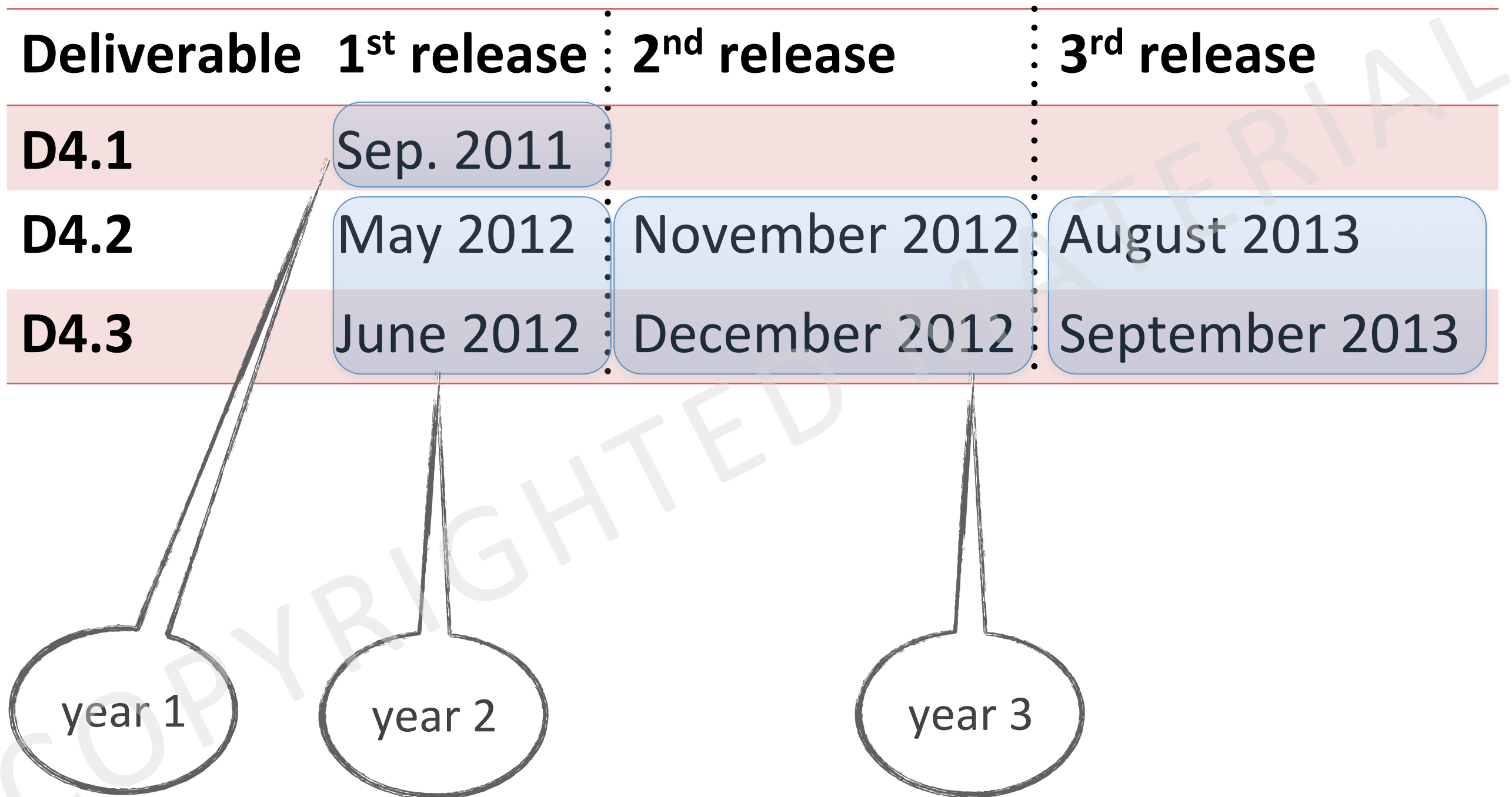
and its deliverables

Deliverable	1 st release	2 nd release	3 rd release
D4.1	Sep. 2011		
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

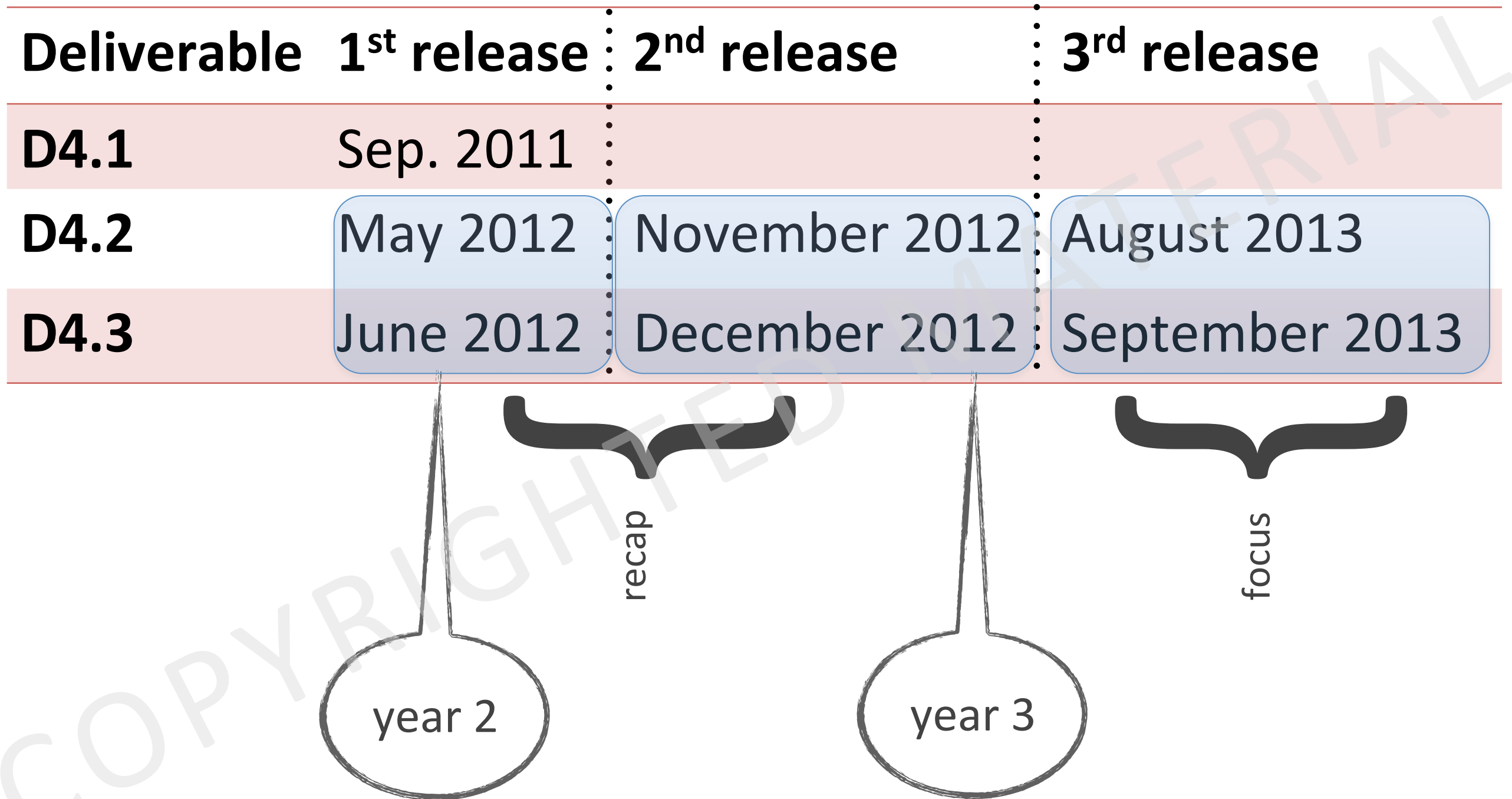
year 1

year 2

and its deliverables



and its deliverables



release 1

Automated Reasoning (AR)

about stories:

- semantics,
- consistency checking
- **deduction** of non-local TLINKs

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

release 1

Automated Reasoning (AR)

about stories:

- semantics,
- consistency checking
- **deduction** of non-local TLINKs

Requirements for	Description
Difficulty levels	Macro levels for learners: <ul style="list-style-type: none"> - entry: character games; - intermediate: time games;
Scheduling of reading and playing	1st silent reading ; 2nd playing smart games; 3rd playing relaxing games
Constraints on actions	Learners should get faster, hence a game has a maximal resolution time
Progress and feedback	Monitor and give learners (1) idea of progress , (2) explanatory feedback, (3) recall their attention and solicit their active resolution (if time)
Representation	Production can be impaired hence promote resolution via visual representation and reasoning

Instructions	Questions	Motivational	Interaction
Choices	Choices for learner		Fixed event
Solutions	Choices or their combination, what are correct/wrong (c/w) solutions		
Feedback	Interaction	Consistency (c/w)	Explanatory Solution
Smart points	Proportional to the learner's ability in the game level		
Relaxing points	Constant		
Avatar	Happy/sad states		
Time	solution constant		interaction constant
Rules	States of the system, actions of the learner, constraints		

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

AR generation of textual games:

- data structure design from game **frameworks** per level (from learner requirements)

release 1

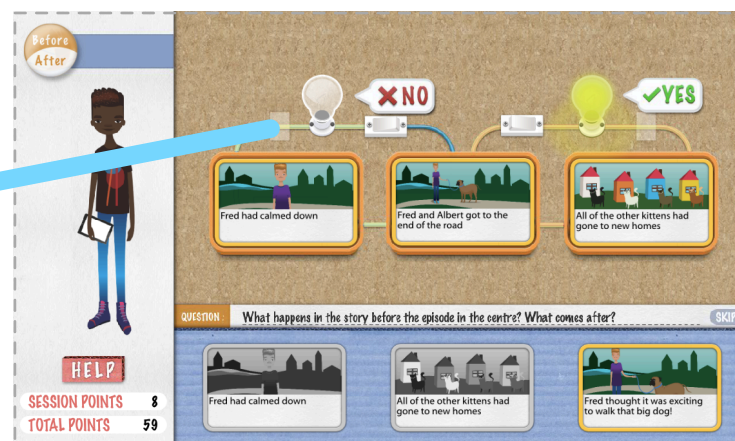
Automated Reasoning (AR)

about stories:

- semantics,
- consistency checking
- **deduction** of non-local TLINKs

Requirements for	Description
Difficulty levels	Macro levels for learners: - entry : character games; - intermediate : time games; - advanced : complex games
Scheduling of reading and playing	1st silent reading ; 2nd playing smart games; 3rd playing relaxing games
Constraints on actions	Learners should get faster, hence a game has a maximal resolution time
Progress and feedback	Monitor and give learners (1) idea of progress , (2) explanatory feedback, (3) recall their attention and solicit their active resolution (if time)
Representation	Production can be impaired hence promote resolution via visual representation and reasoning

Instructions	Questions	Motivational	Interaction
Choices	Choices for learner		Fixed event
Solutions	Choices or their combinations that are correct/wrong (c/w) solutions		
Feedback	Interaction	Consistency (c/w) explanatory	Solution
Smart points	Proportional to the learner's ability in the game level		
Relaxing points	Constant		
Avatar	Happy/sad states		
Time	solution constant		interaction constant
Rules	States of the system, actions of the learner, constraints		



Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

AR generation of textual games:

- data structure design from game **frameworks** per level (from learner requirements)
- generation of main data, e.g., central events and solutions, with **generate-and-test**

release 2

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

-**AR generation** of central events and solutions: from generate-and-test to **generate-with-constraints**

Algorithm 2 The before-after games generation algorithm

Require: event e , story s , story $s_o \neq s$

```

1: foreach tlink  $t_1$  in  $s$ , that has  $e$  as target do
2:   foreach tlink  $t_2$  in  $s$ , that has  $e$  as source do
3:     select a random event  $w$  from story  $s_o$ 
4:     create a before/after game that has:
5:        $e$  as fixed event
6:       the source event of  $t_1$  as before
7:       the target event of  $t_2$  as after
8:        $w$  as wrong event
9:   end for
10: end for

```

from s so that
 w includes/
included_by e

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

- **AR generation** of central events and solutions: from generate-and-test to **generate-with-constraints**
- **Natural language generation** components for **EN** and **IT**

Algorithm 2 The before-after games generation algorithm

Require: event e , story s , story $s_o \neq s$

```

1: foreach tlink  $t_1$  in  $s$ , that has  $e$  as target do
2:   foreach tlink  $t_2$  in  $s$ , that has  $e$  as source do
3:     select a random event  $w$  from story  $s_o$ 
4:     create a before/after game that has:
5:        $e$  as fixed event
6:       the source event of  $t_1$  as before
7:       the target event of  $t_2$  as after
8:        $w$  as wrong event
9:   end for
10: end for
  
```

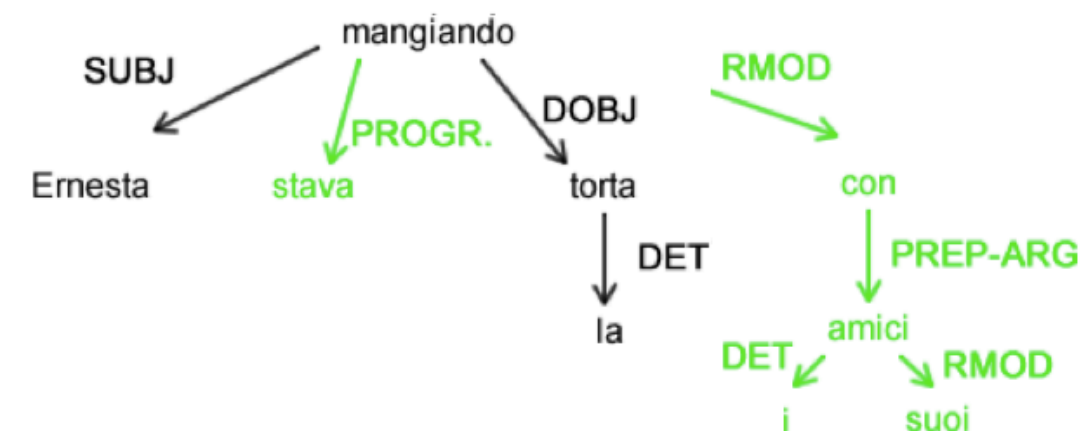
from s so that
 w includes/
included_by e

INPUT: Luke **raced**(event) **past** him and **reached** other end first.

(ROOT
(S
(NP (NNP Luke)) ← **Subject** (dependency info: N_subj)
(VP
(VP (VBD **raced**) → **verb = race, aspect = "simple"**
(PP (IN past)) ← **Context** (VP sub tree)
(NP (PRP him)))) → **"him" refers_to Ben (Coreference)**
(CC and)
(VP (VBD reached)
(NP (DT other) (NN end)) ← **Redundant information**
(ADVP (RB first))))
(. .)))

OUTPUT: Luke races past Ben / Who races past Ben?

INPUT: Ernesta stava mangiando la torta con i suoi amici.



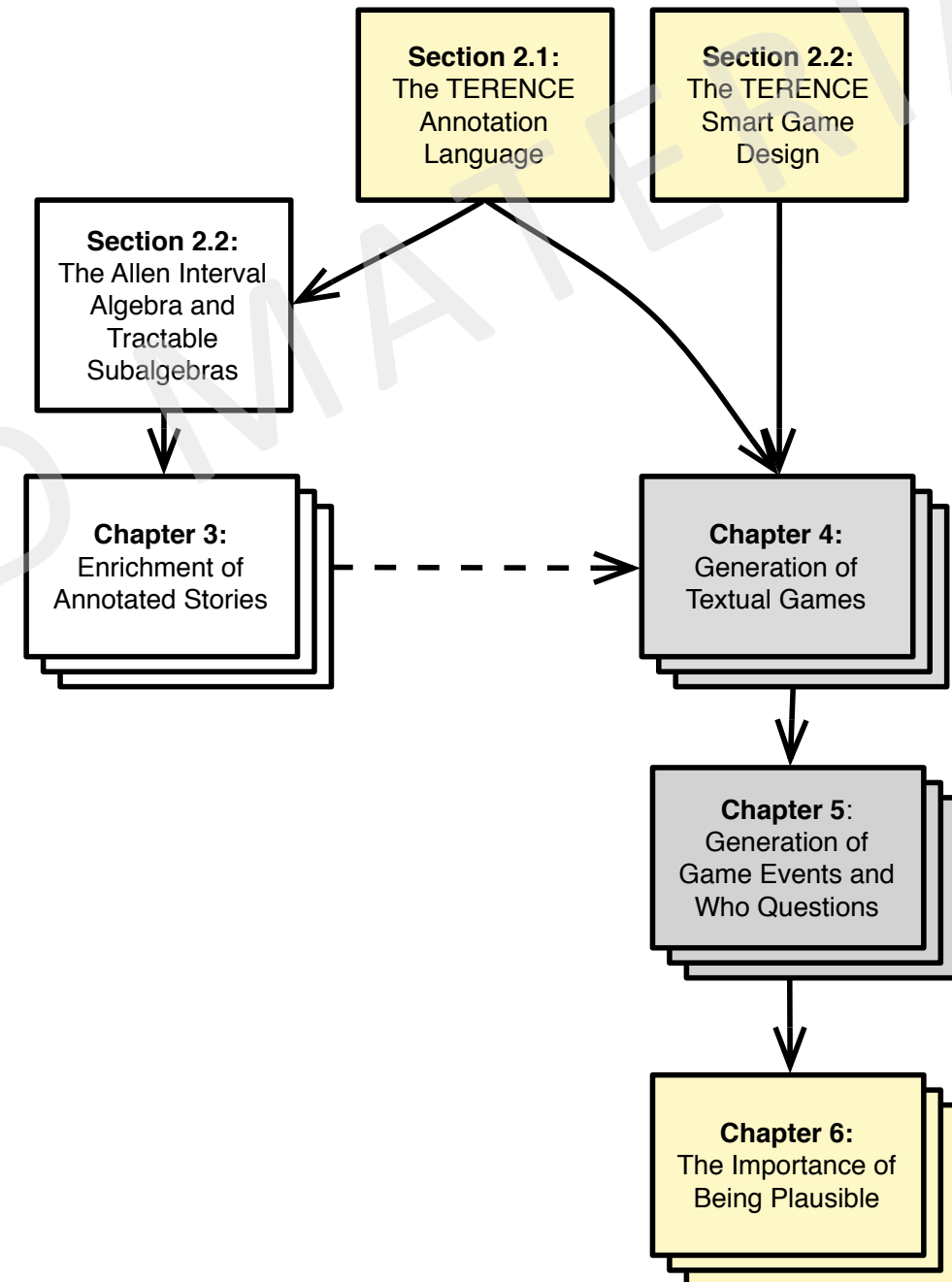
OUTPUT: Ernesta mangia la torta.

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

Action 1: code optimisation for the generation workflow

Action 2: analysis of impact of WP7 evaluation, design revisions for generation components

Action 3: documenting of the workflow in technical annex to D4.2 and D4.3 and paper writing



From D4.2 and D4.3 technical annex

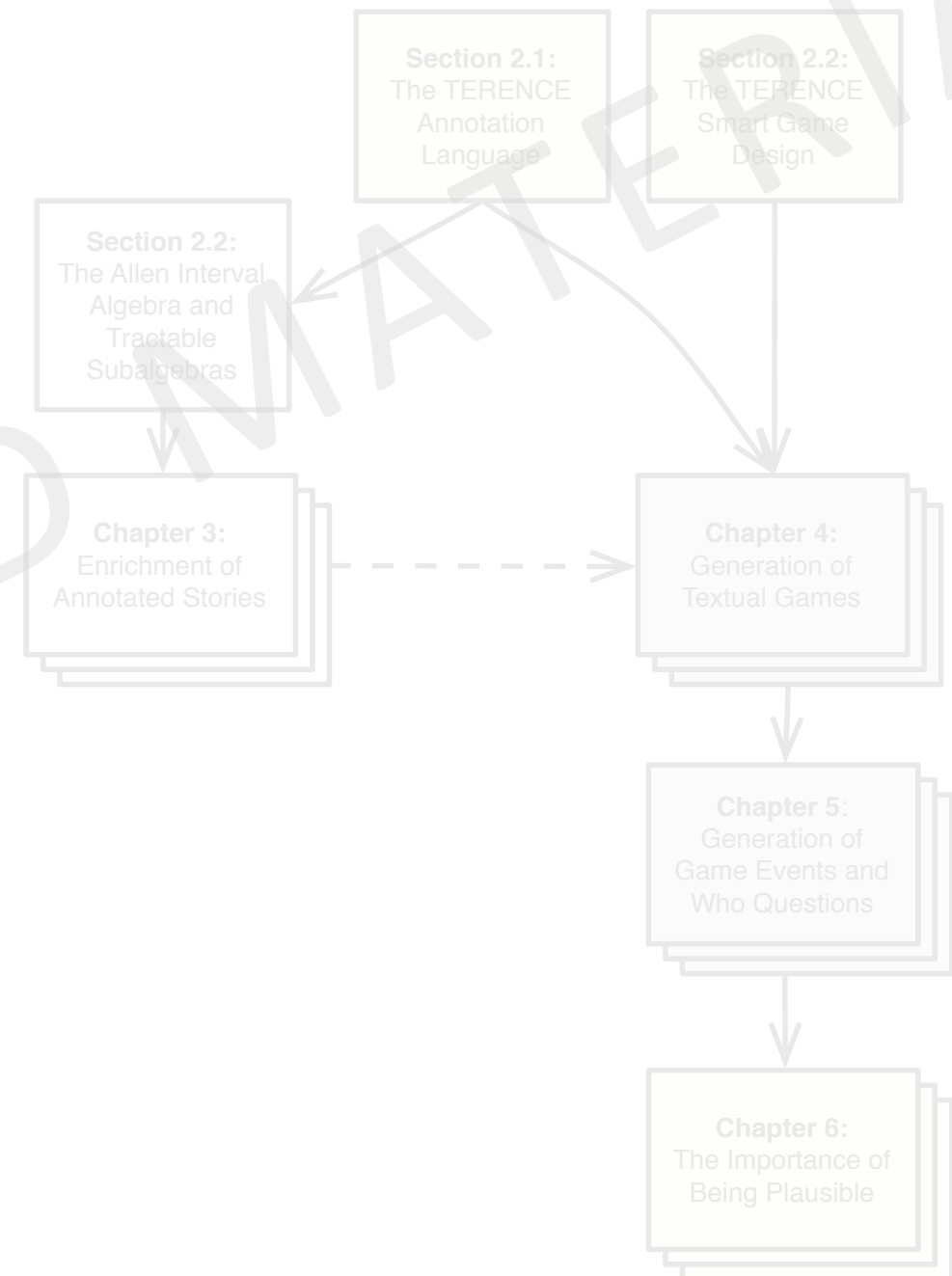
release 3

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

Action 1: code optimisation for the generation workflow

Action 2: analysis of impact of WP7 evaluation, design revisions for generation components

Action 3: documenting of the workflow in technical annex to D4.2 and D4.3 and paper writing



From D4.2 and D4.3 technical annex

Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

Action 1: code optimisation for the generation workflow, e.g.,

-for reasoning about stories,

- ▶ hard consistency checking

- ▶ relaxed consistency checking

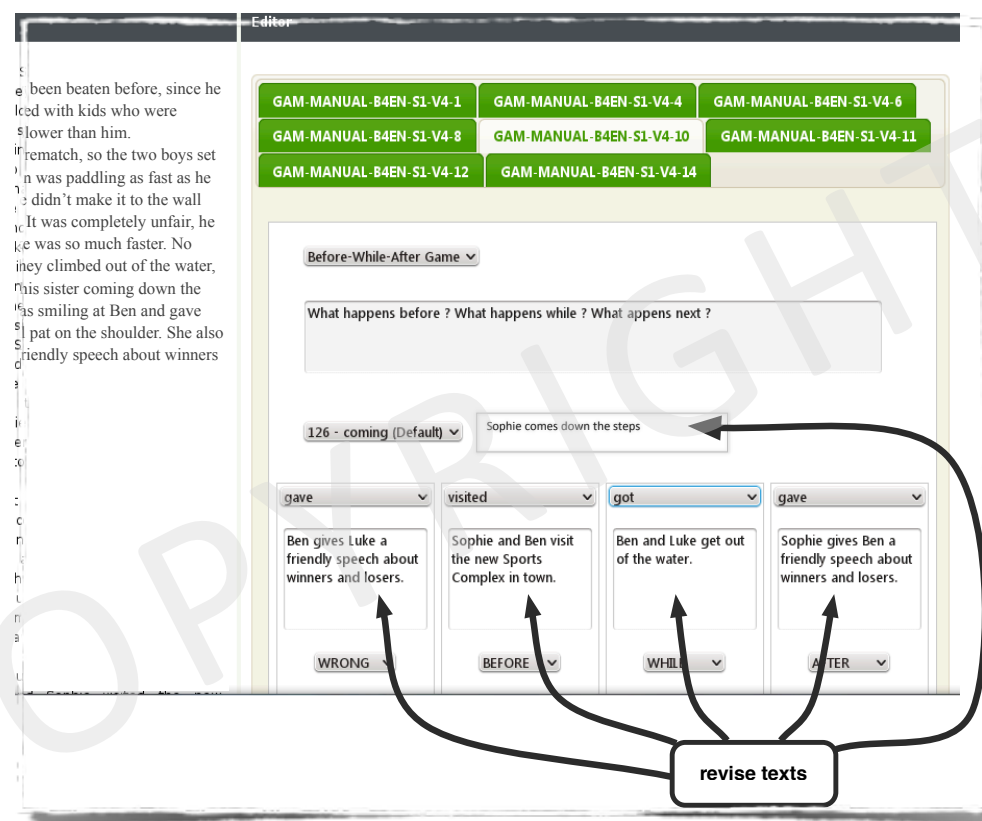
-for natural language generation components

From D4.2 and D4.3 technical annex

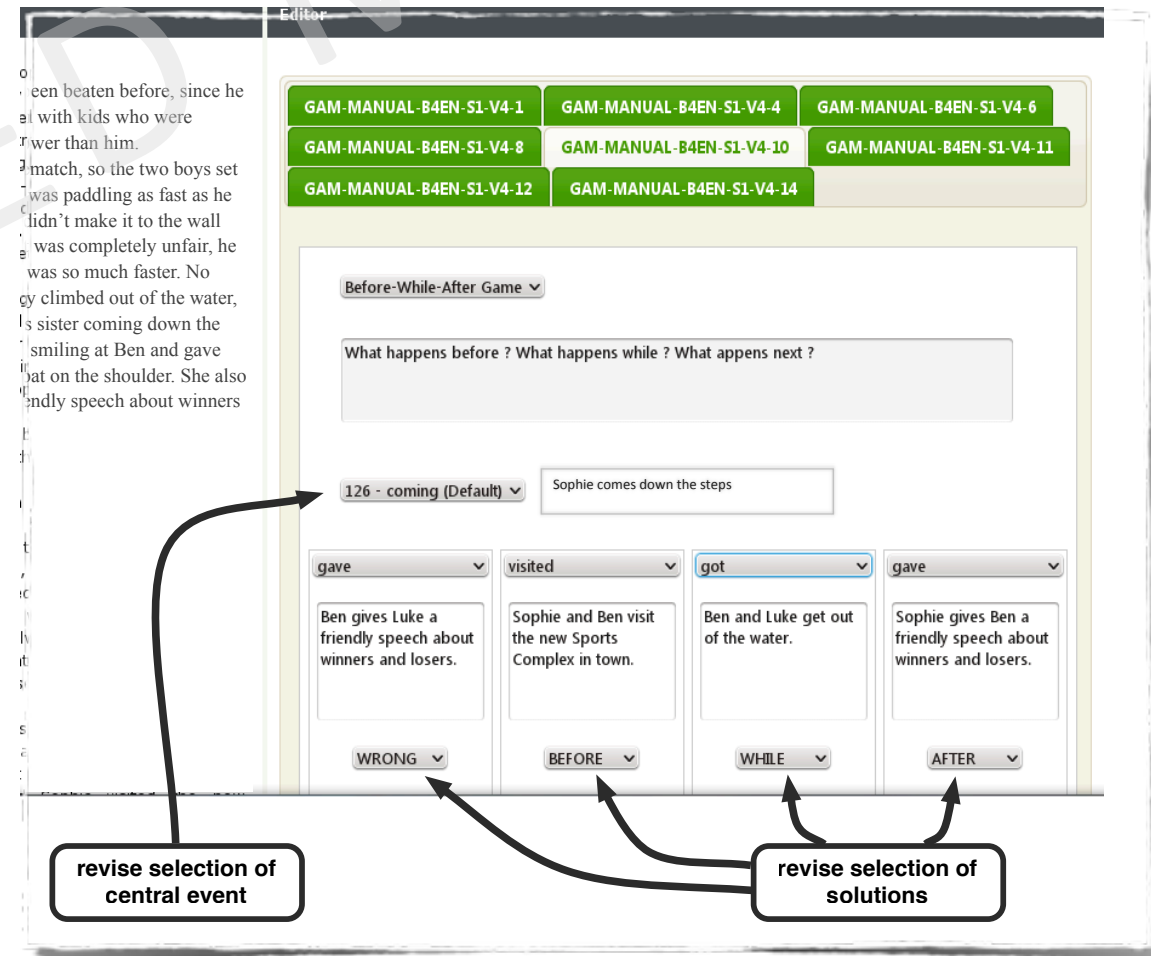
Deliverable	1 st release	2 nd release	3 rd release
D4.2	May 2012	November 2012	August 2013
D4.3	June 2012	December 2012	September 2013

Action 2: (1) analysis of WP7 expert-based evaluation results, (2) design revisions (Y2-Y3). E.g.,

- for selection of central events and solutions



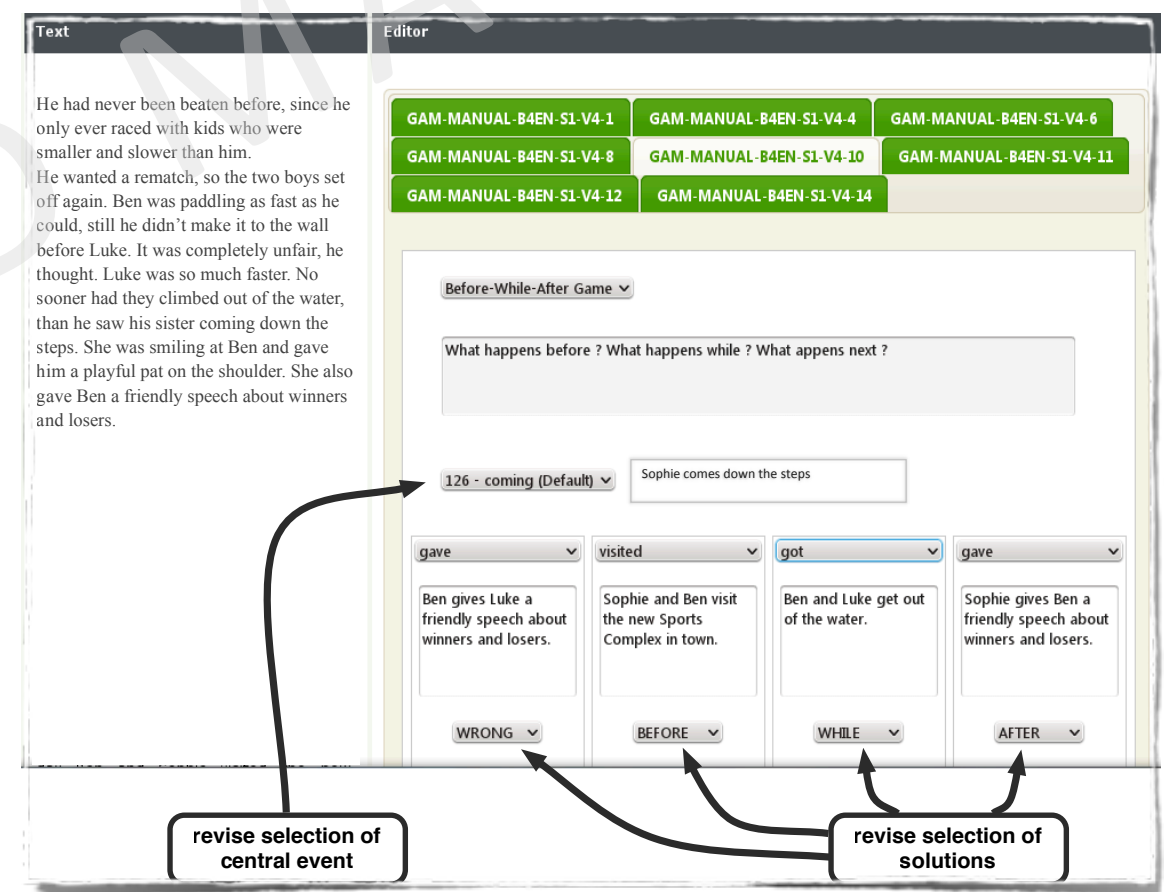
From D4.2 and D4.3 technical annex



release 3: action 2

Selection of central events for games:

- > *Results*: only in 15 out of 250 cases (6%), it was necessary to select a different central event than the automatically generated one
- > *Implications for WP4*: none picked up



From D4.2 and D4.3 technical annex

release 3: action 2

Selection of plausible solutions:

- > *Results:* out of 140 changes of selection of solutions, the majority was for wrong solutions
- > *Implications for WP4:* WP4 mainly worked on optimising modules and developing new heuristics for **wrong plausible solutions** in the last part of Y3,

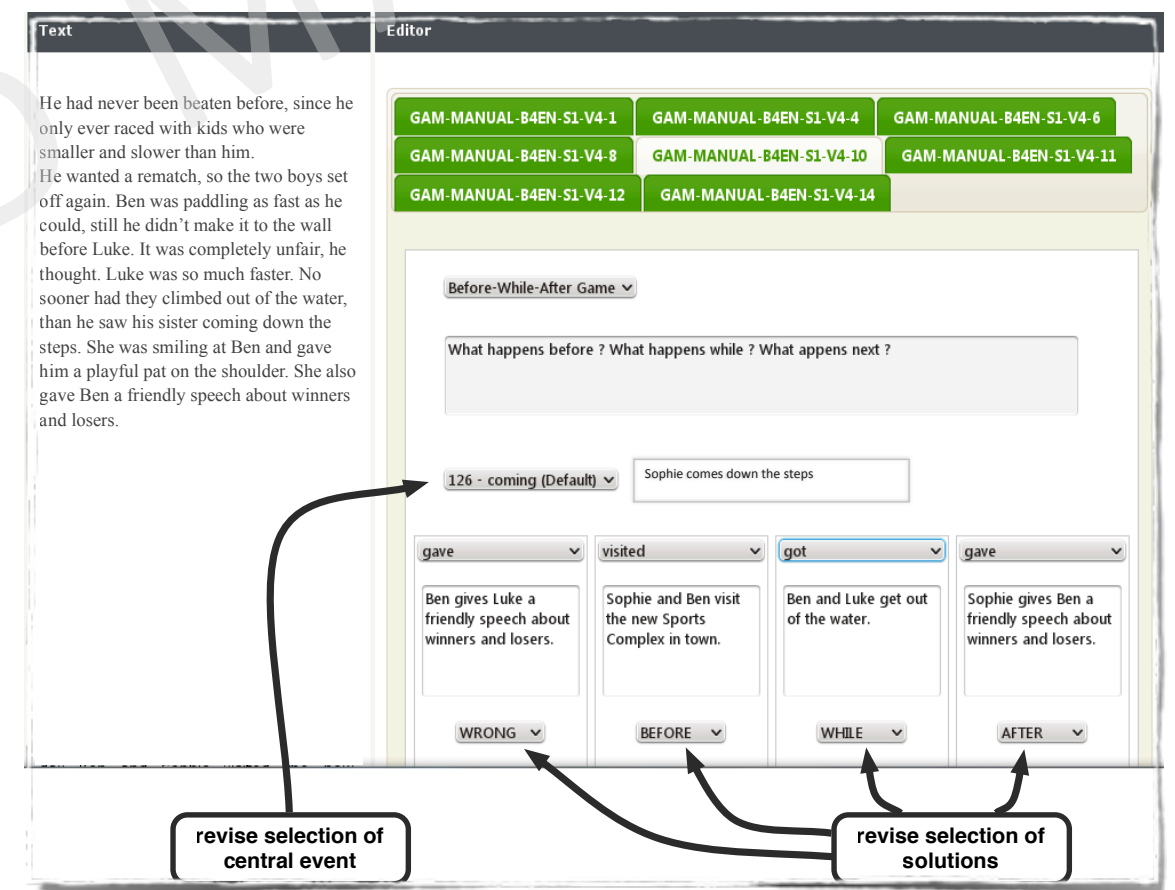
- generate a wrong solution from correct one by changing participants, e.g.,

```
<correct_sentence id="2">
Sul razzo, viaggiano Ernesta e
la cagnolina Chiazza.
```

```
</correct_sentence>
```

```
<wrong_sentence id="2wh1" >
Sul razzo, viaggiano Ernesta e
Nerina.
```

```
</wrong_sentence>
```



From D4.2 and D4.3 technical annex

release 3: action 2

Development times:

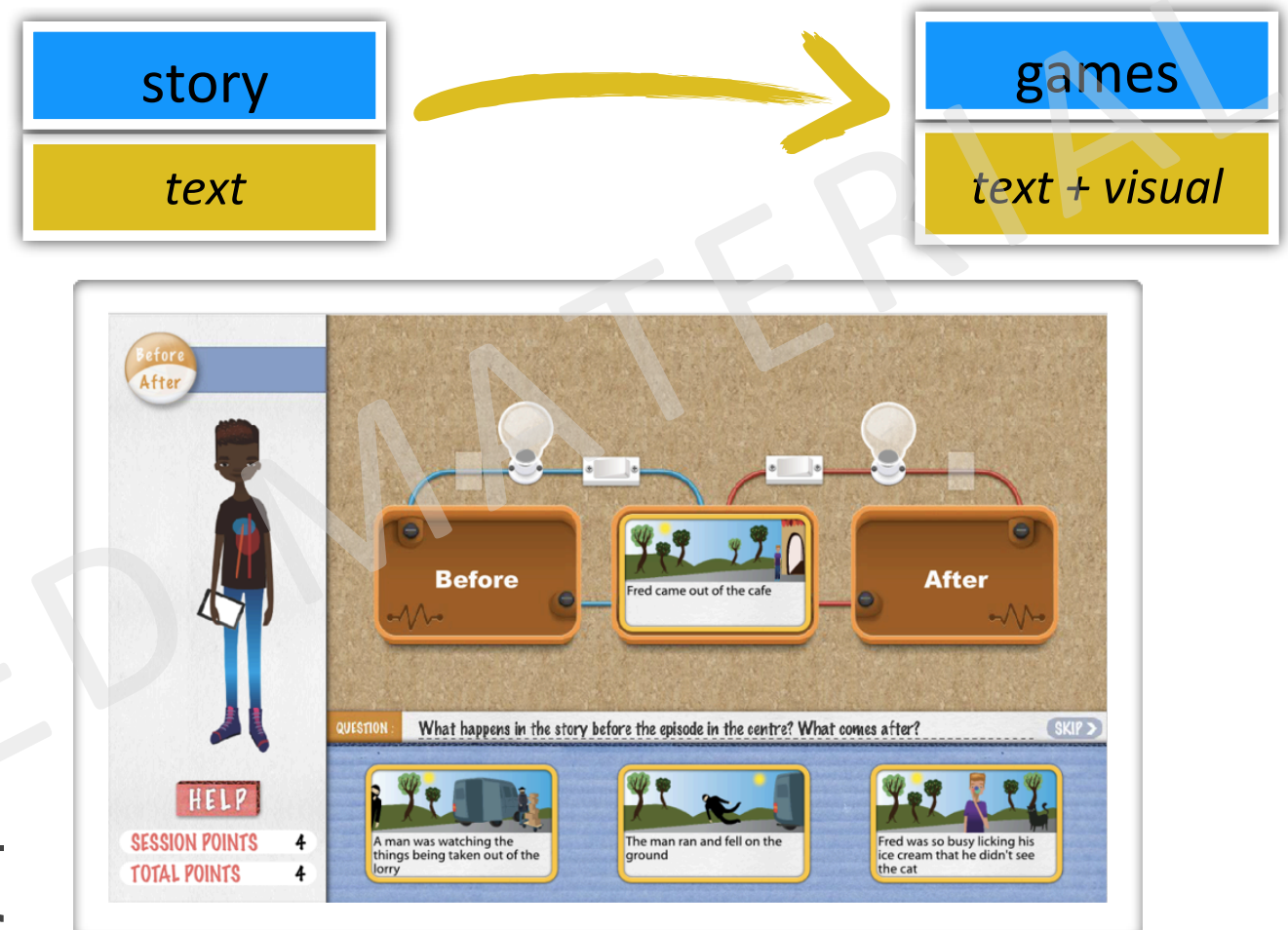
>Results for revision time:

- **12,6** m. per game instance
- ↑ times: 12,8 m. for time games
- ↓ time: 10,6 m. for who games

>Results for creation time:

- avg. **23** m. per game instance

>Implications for WP4: the semi-automated development process is promising for optimising development times



From D4.2 and D4.3 technical annex

game over

Deliverable 1st release 2nd release 3rd release

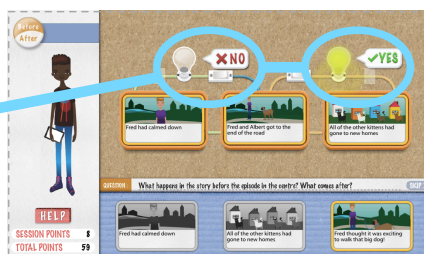
D4.1 Sep. 2011

D4.2 May 2012 November 2012 August 2013

D4.3 June 2012 December 2012 September 2013

Requirements for	Description
Difficulty levels	Macro levels for learners: - entry : character games; - intermediate : time games; - advanced : complex games.
Scheduling of reading and playing	1st silent reading ; 2nd playing smart games; 3rd playing relaxing games.
Constraints on actions	Learners should get faster, hence a game has a maximal resolution time .
Progress and feedback	Monitor and give learners (1) idea of progress , (2) explanatory feedback, (3) recall their attention and solicit their resolution (if/when).
Representation	Production can be impaired hence promote resolution via visual representation and reasoning .

Instructions	Questions	Motivational	Interaction
Choices	Choices for learner		Fixed event
Solutions	Choices or their combination: what are correct/wrong (c/w) solutions		
Feedback	Interaction	Consistency (c/w)	explanatory
Smart points	Proportional to the learner's ability in the game level		
Relaxing points	Constant		
Avatar	Happy/sad states		
Time	solution constant		interaction constant
Rules	States of the system, actions of the learner, constraints		



Algorithm 2 The before-after games generation algorithm

Require: event e , story s , story $s_o \neq s$

```

1: foreach link  $t_1$  in  $s$ , that has  $e$  as target do
2:   foreach link  $t_2$  in  $s$ , that has  $e$  as source do
3:     select a random event  $w$  from story  $s_o$ 
4:     create a before/after game that has:
5:        $e$  as fixed event
6:       the source event of  $t_1$  as before
7:       the target event of  $t_2$  as after
8:        $w$  as wrong event
9:   end for
10: end for
  
```

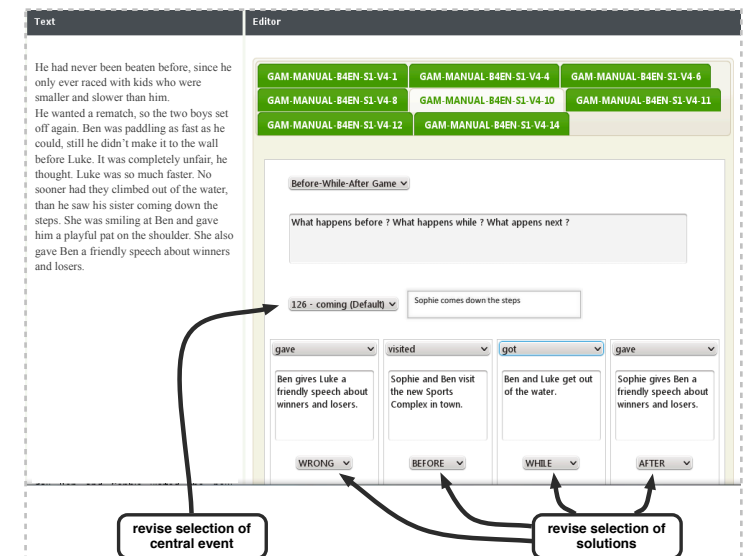
from s so that
 w includes/
included_by e

INPUT: Luke **raced**(event) **past** him and **reached** other end first.

(ROOT
(S
(NP (NNP Luke)) ← **Subject** (dependency info: N_subj)
(VP
(VP (VBD **raced**) ← **verb = race, aspect = "simple"**
(PP (IN past) ← **Context** (VP sub tree)
(NP (PRP him)))) ← **"him" refers_to Ben (Coreference)**

(CC and)
(VP (VBD **reached**)
(NP (DT other) (NN end)) ← **Redundant information**
(ADVP (RB first))))
(. .)))

OUTPUT: Luke races past Ben / Who races past Ben?



game over ?

Deliverable 1st release 2nd release 3rd release

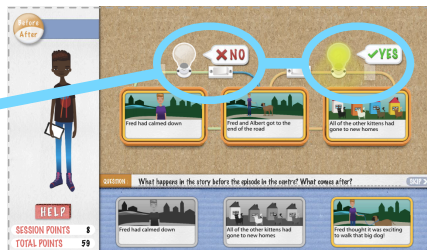
D4.1 Sep. 2011

D4.2 May 2012 November 2012 August 2013

D4.3 June 2012 December 2012 September 2013

Requirements for	Description
Difficulty levels	Macro levels for learners: - entry: character games; - intermediate: time games; - advanced: complex games.
Scheduling of reading and playing	1st silent reading; 2nd playing smart games; 3rd playing relaxing games
Constraints on actions	Learners should get faster, hence a game has a maximal resolution time
Progress and feedback	Monitor and give learners (1) idea of progress, (2) explanatory feedback, (3) recall their attention and solicit their resolution (if time)
Representation	Production can be impaired hence promote resolution via visual representation and reasoning

Instructions	Questions	Motivational	Interaction
Choices	Choices for learner		Fixed event
Solutions	Choices or their combination: what are correct/wrong (c/w) solutions		
Feedback	Interaction	Consistency (c/w)	Explanatory
Smart points	Proportional to the learner's ability in the game level		
Relaxing points	Constant		
Avatar	Happy/sad states		
Time	solution constant		interaction constant
Rules	States of the system, actions of the learner, constraints		



Algorithm 2 The before-after games generation algorithm

Require: event e , story s , story $s_o \neq s$

```

1: foreach link  $t_1$  in  $s$ , that has  $e$  as target do
2:   foreach link  $t_2$  in  $s$ , that has  $e$  as source do
3:     select a random event  $w$  from story  $s_o$ 
4:     create a before/after game that has:
5:        $e$  as fixed event
6:       the source event of  $t_1$  as before
7:       the target event of  $t_2$  as after
8:        $w$  as wrong event
9:   end for
10: end for

```

from s so that w includes/ included_by e

INPUT: Luke raced(event) past him and reached other end first.

```

(ROOT
(S
(NP (NNP Luke)) ← Subject (dependency info: N_subj)
(VP
(VP (VBD raced) ← verb = race, aspect = "simple"
(PP (IN past) ← Context (VP sub tree)
(NP (PRP him)))) ← "him" refers_to Ben (Coreference)
(CC and)
(VP (VBD reached)
(NP (DT other) (NN end)) ← Redundant information
(ADVP (RB first))))
(. .)))

```

OUTPUT: Luke races past Ben / Who races past Ben?

