

# A Formalism for Temporal Annotation and Reasoning of Complex Events in Natural Language

Francesco Mele<sup>1</sup>, Antonio Sorgente<sup>1</sup>,

<sup>1</sup> Istituto di Cibernetica “E. Caianiello”, Consiglio Nazionale delle Ricerche  
Via Campi Flegrei, 34 Pozzuoli(Napoli) Italia  
{f.mele, a.sorgente}@cib.na.cnr.it

**Abstract.** This work has been developed with the motivation of defining a formalism (OntoTimeFL) for annotating complex events in natural language texts and applying to the items annotated several types of axioms and rules for temporal reasoning. In part, OntoTimeFL is a conceptualization of TimeML formalism, where the basic concepts of annotation by an ontological form have been represented. In addition, OntoTimeFL introduces new constructs for the annotation that mainly concern three complex events: narrative, intentional, and causal. OntoTimeFL entities have been defined as classes of a formal ontology; a methodological choice for facilitating automatic annotation processes to reuse existing axiomatics in the research of temporal reasoning, and facilitate the creation of new ones (some proposals in this paper).

**Keywords:** Causal reasoning, Complex Event, Natural Language, Ontology, Frame Logic

## 1 Introduction

One of the main goals of this research is to build a model and a formalism for the representation and reasoning of complex events in Natural Language (NL).

In this context, attention to the intentional, causal, and narrative entities present in NL have been focused on. In NL research, many theories concerning the events have been produced, in particular for the formalization of a narrative that has been considered as a set of entities (events) and a set of temporal order relations between these events. In Artificial Intelligence (AI) much research on Mental Attitudes [COU] (Desire, Belief, Intention), has been developed. Furthermore, in AI, Logic, and Philosophy [TOO], causality has been studied as a relationship between events and a set of rules (axioms) that justify and/or predict the causal order between events. In this paper, in the formalism proposed (OntoTimeFL), intentions, causations, and narratives, have been considered as complex events. Those entities can be themselves components of other complex events. For this purpose, a representation of complex events has been defined, where each complex event has:

- an interval and a temporal mode (before, during, etc. with respect to another interval, typical of simple events);

- an explicit representation by an ontological formalization, and by specific rules of composition for each type of complex event (causal, mental, etc.); and,
- a specific axiomatic associated to each class (rules of inference that can only be applied, for example, to causal or intentional events).

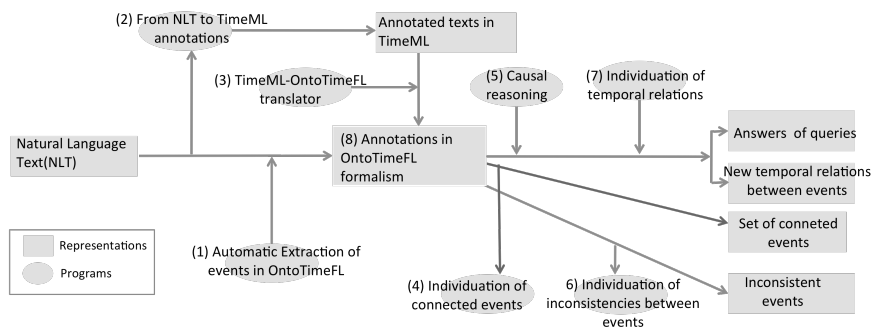
The OntoTimeFL formalism permits the annotation of complex events like:

(Io<sub>a1</sub> credevo<sub>e1</sub> che tu<sub>a2</sub> non avessi voglia<sub>act2</sub> di andare<sub>e3</sub> al cinema ecco perché ti ho invitato<sub>e4</sub> a cena)<sub>e6</sub>

(I<sub>a1</sub> believed<sub>e1</sub> that you<sub>a2</sub> did not want<sub>act2</sub> to go<sub>e3</sub> to the cinema, that's why I invited<sub>e4</sub> you to dinner)<sub>e6</sub>

Informally, OntoTimeFL defines a complex intentional event (Io<sub>a1</sub> credevo<sub>e1</sub> che tu<sub>a2</sub> non avessi voglia<sub>act2</sub> di andare<sub>e3</sub> al cinema)<sub>e5</sub> that is composed of a mental event e1, which has as its argument in another mental event act2 (volere), which itself has another argument event (e3). In addition, the event e5 is represented as an event that happens over time, and, therefore, is a component of the causal event e6 (characterized by the causal relation *cause*(e5, e4)).

With this representation then, one can annotate causal relations and associate rules (axioms) for their causal reasoning. In OntoTimeFL, these axioms can be easily integrated with other axioms of temporal reasoning by a well-known rule: if, *cause*(e1, e2) then, *precedes*(e1, e2). In recent years, the formalism TimeML[CAS, SAU, PUS] for annotating events in natural language has emerged. Around this formalism, several research projects, which have produced many data banks of text annotations using the TimeML formalism [COR], have been born. OntoTimeFL permits all annotation types derived from TimeML, as previously reported, for annotating complex events. For this reason, a software module has been developed, which takes existing annotations in TimeML and translates them into OntoTimeFL (module (3) of Fig 1). This program was inserted in order to render compliant OntoTimeFl with TimeML and recover the wide set of annotations in TimeML.



**Fig.1:** Modules for the temporal reasoning

The diagram (see Fig. 1) represented all modules considered (at present implemented or yet to develop). In this paper, the OntoTimeFL formalism (8) and the Casual Reasoning module(5) have been presented. Starting from OntoTimeFL formalism is possible to associate several axiomatics regarding: the detection of

inconsistencies in events (6), the discovery of new temporal relations (7), and the analysis of their connectivity (4) (for more details about module (4), (6), and (7) see reference [MEL]).

This research has for its goal the creation of an architecture that allows, in completely automatic way, the processing of text in natural language (eventually) through a TimeML annotations, and the application of complex functionalities for temporal reasoning.

For the module (2), an automated annotation that performs partial annotation (annotated events only) in TimeML of natural language texts [ROB] was produced.

## **Related Work**

The TimeML is a standard mark-up language for annotating events in a document. TimeML resolves some problems regarding the annotation of events; including how to represent the anchorage of events on the temporal axis, the relations in which events have with respect to one another, and the attempted detection of temporal references when the context of the temporal expressions is underspecified. Some research projects about several natural languages have been developed in which some wide data banks have been provided.

Another important research project relating to the annotation of linguistic expressions is LMF (Lexical Markup Framework) [LMF], by which structures of the lexicon can be defined, or more precisely: LMF represents linguistic entities and their usage, through classes, their instances, and class relations associating such structures with the English text and using UML diagrams for the visualisation. Although UML has the "class" as main entity of representation, it is not a computational apparatus. For this reason, UML does not permit (directly, and in the same formalism) the association of axiomatics to classes. In other words, UML is not an ontological formalism.

Recently, in latter the direction new research has been born. In [BUI] a unified model for associating linguistic information to ontologies was presented. In this approach there are two models LingInfo and LexOnto. The LingInfo model provides a mechanism for modelling the linguistic structures, while the LexOnto model enables the representation of external linguistic structures ( predicate-argument structures in the form of ontological terms) and their association with to corresponding ontological elements. This work also provides some requirements for associating linguistic information with ontologies; one of these consists of requiring representations in which the Linguistic and Ontological Levels are separated. LexInfo is based on the LMF formalism.

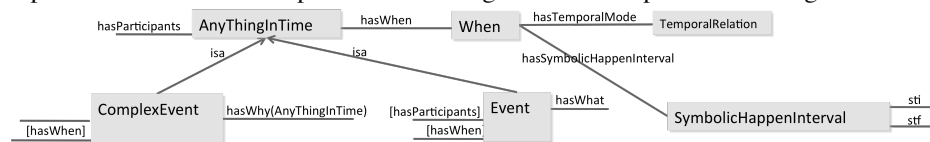
In other recent research [WIN], there have been several proposals for representing events. The basic motivation of this research consists in the belief that the events can constitute an excellent framework for aggregating knowledge. The large quantity of data and knowledge (fragmented and unstructured) on the Internet, makes this research very attractive. An emerging methodology for the representation of events knowledge distributed on the Internet has been named Event-Centric [WIN].

To represent events, some formalisms have been inspired from a model that has its roots in journalism. This model called "W's and one H " adopts six attributes for the

representation of events: Who, When, Where, What, Why, and How. The project Eventory [WAN] adopts the model "W's and one H".

## 2 The representation of events

In this work, an ontology for complex events has been defined: `OntoTimeFL`. `OntoTimeFL` has an abstract superclass (`AnythingInTime`) common to all entities that happen over time. Two subclasses are specializations of `AnythingInTime`: `Event`, which represents the class of simple events, and `ComplexEvent`, which represents the class of complex events. In Fig. 2 a sketch representation is given:



**Fig. 2.** Taxonomy of the classes of simple and complex events.

In Fig.2, in brackets, the attributes that are inherited from their respective superclasses are reported. `AnythingInTime` is an abstract class (without instances) which is the superclass of the concrete classes: `Event` and `ComplexEvent`. The latter classes are the key concepts of the formalism `OntoTimeFL`. The `Event` class has the descriptor `hasWhat`, which is associated with the class `Action`. Generally, this class describes an action (which happens over time) that characterizes an event or describes a property that is true in a specific time interval. The `ComplexEvent` is an abstract class and represents complex events through a set of events (components), and of relations between events. Its subclasses (see par. 2.4), in accordance with the type of relationship that exists between components, have been classified. `ComplexEvent` also has the attribute `hasWhy` that describes the causal relations between events. `Why` is a relation between two events; for this reason, it can only be the attribute of a complex event.

### 2.1 Instant and interval representations

The representation of time that has been adopted is mixed; ie, it is based on points and time intervals. In `OntoTimeFL`, all temporal entities are represented as classes (Fig. 3). `TemporalTerms` is the main class and has several specializations: date or partial date (`DateValue`), time instants or combinations of them with date (`TimeValue`), symbolic times (`Symbolic`), and time intervals (`IntervalValue`). In addition, there are classes for annotating temporal entities in the text (Fig. 3). The definition of these classes has been inspired by the formalism `TimeML`. The root of the taxonomy is `TimeX3`, and its subclasses are: `TIMEX3Data`, `TIMEX3Time`, `TIMEX3Set`, and `TIMEX3Duration`. These classes are associated with the `TemporalTerms` class.

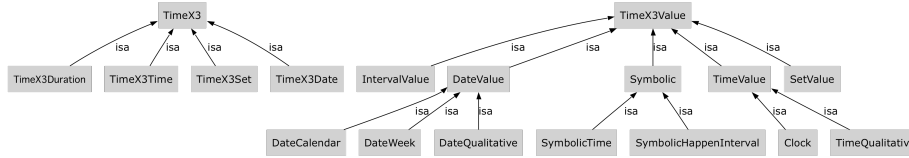


Fig 3. Taxonomy of temporal terms

## 2.2 The descriptor When

In OntoTimeFL there is a particular structure: the class `When` (see Fig. 2), which describes when an event happens using the effective symbolic interval (ESI) in which the event happens, and a temporal modality of happening, described by one (or more) temporal order relations (before, after, during, etc.) between ESI and some temporal interval of reference (or also another event). These relations have the goal of anchoring an event to the chronological axis, or with another event through a temporal order relation. The approach requires, therefore, that when an event (simple or complex) is created, it automatically generates a type identifier ESI, represented by two attributes: `sti`, the (effective) symbolic time in which the event starts (or in which the property is true), and `stf`, the time in which the event ends. The choice of having an effective time when an event happens and a temporal modality of happening, is motivated by the fact that often the effective time in which an event happens is not known, but one can easily know one or more relations for it(modality of happening/ after a time `tx/ dtx` before a certain range, etc.). Thus, even if one does not exactly know the exact value of the start and/or end of an event, one can annotate (or, automatically discover) relationships with other time intervals, as soon as they become available. The class `When` has the following definition:

```

When [
  hasSymbolicHappenInterval*=>SymbolicHappenInterval,
  hasTemporalMode*=>TemporalRelation].
SymbolicHappenInterval[sti*=> SymbolicTime,
  stf*=> SymbolicTime].
  
```

## 2.3 Simple Events

In OntoTimeFL, the class `Event` represents simple events. This class inherits the attributes of the superclass `AnythingInTime` and has the attribute `hasWhat`, which describes an action that happens or a property that is true in a temporal interval.

```

Event[hasWhat*=>Action].
  
```

The attribute `hasWhat` has values in the `Action` domain and it describes exactly what happens (action) or what is true (property) in a temporal interval.

The formalism has an explicit definition of the "action" (by the class `action`). The subclasses of `Action` have been defined in accordance with the POS (part of speech): `AdjectiveAction`, `NounAction`, `PrepositionAction`, and `VerbAction`. Each subclass has subcategories inspired by TimeML formalism like `Reporting`, `Occurrence`, `Perception`, `State`, etc. Furthermore, `Mental Acts` (mental attitude) have been added compared to TimeML.

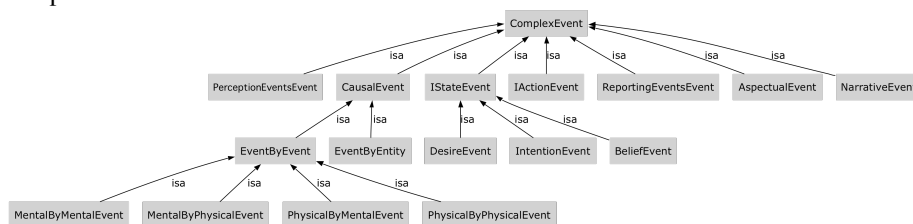
## 2.4 Complex Events – Taxonomy and Composition

The `ComplexEvent` class is described by the method `hasWhy(AnythingInThing)`, a function, that given an input event belonging to a complex event, returns a set of causal relations that are the justification of why the event occurred.

`ComplexEvent[hasWhy(AnythingInThing)*=>CausalRelation]`.

Complex events are defined by a temporal mode, described by the descriptor `hasWhen`, the same attribute used for the description of simple events. For the simple event, the `hasSymbolicHappenInterval` term defines the temporal interval in which the action happens; while the time interval of occurrence of the complex event defines the minimum interval of time in which all events belong to the complex event that occurs. Therefore, the occurrence interval of complex events is not continuous, or also, not in all temporal subintervals is there an event that happens. The descriptor `When` can be calculated in according to the descriptors of the events' components, or it is instantiated interactively. In the latter case, compatibility checks (defined by constraints), with respect to the attributes `When` of the events' components, must be run.

Fig. 4 shows the taxonomy of complex events of the `OntoTimeFL` ontology, where the narrative events, the causal events, the intentional events (subclasses of `IStateEvent`), and the perceptual events have been labeled and represented as complex events.



**Fig.4** The taxonomy (subclasses) of complex events.

### Narrative Events

Complex narrative events (`NarrativeEvent`) define an aggregation of events connected by temporal relations. The components of `NarrativeEvent` can be simple events that describe actions that occur over time, or properties that are true in a

temporal interval, or other complex events such as causal events (*CausalEvent*) or intentional events (*IStateEvent*).

*NarrativeEvent*, like all the subclasses of *AnythingInTime*, inherits the attributes *When* and *Participants*, and like all the subclasses of *ComplexEvent*, inherits the method *hasWhy*(*AnythingInThing*). The characterization of *NarrativeEvent* is given by the attribute *hasTemporalRule*, which defines temporal relations between components of the events. In fig. 5 an example of the annotation of the narrative event is given.

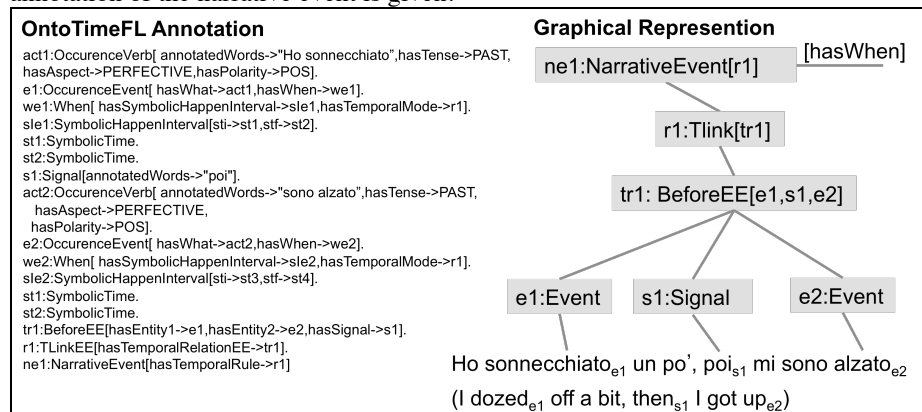


Fig. 5: An example of a narrative event.

### Mental events

Mental events are defined, through the attributes *When* and *Participants* (inherited by *AnythingInTime*), and by a slot that describes a relation (*hasISateRelation*) between a mental event and a physical event. In fig. 6 an example of an annotation of mental event is given:

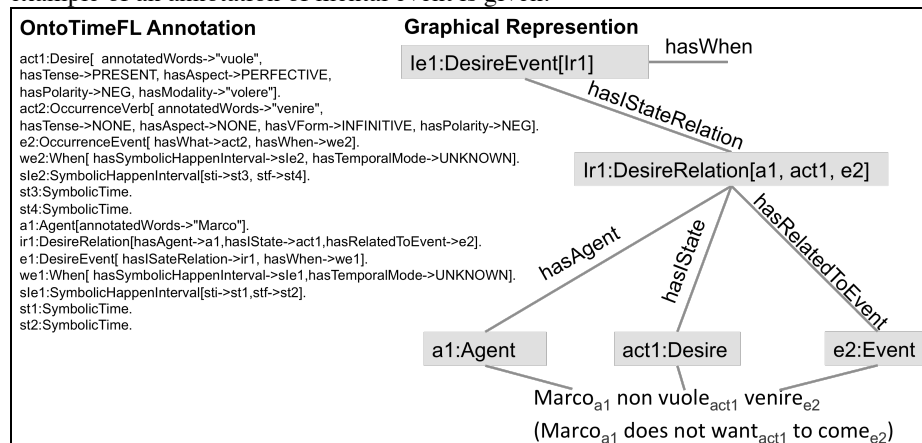


Fig. 6: An example of a mental event.

### Causal events

The complex event `CausalEvent` describes an event that relates to a cause-effect relation: the occurrence of an event (event cause) caused the occurrence of another event (event effect). Causal events are defined by a causal relationship between events, and like all event subclasses of `AnythingInTime`, inherit the attributes `When` and `Participant`. In `OntoTimeFL`, a classification of causal events in accordance with the nature of the events involved has been defined, or, that is, if the cause-effect relationship is defined by physical and/or mental events.

The causal events `MentalByPhysicalEvent` are events formed by mental events (effects) and physical events (cause). The `PhysicalEvent` in these are perception action (see, look, smell, feel, etc.), while mental events are beliefs, desires and intentions. For example: “*He laughed and I thought he was joking*”.

The causal events `PhysicalByMentalEvent` are characterized by a `PhysicalEvent` that one thinks can be caused by a mental state of an agent. For example: “*I think it's a good book, I'll buy it*”, and “*I would like something hot, I'll take a cup of tea*”.

The causal events `PhysicalByPhysicalEvent` are events composed by two events: the physical event cause and the physical event effect. For example: “*He bumped the glass with his elbow and broke it*”, and “*It's raining and the road is wet*”.

The `MentalByMentalEvent` are events composed of two mental events: the mental event cause and the mental event effect. For example: “*I think it's the best team and I think it will win the championship*”.

For causal events, a widely shared relation (axiom), that brings together the causal relations with temporal relations, has been defined:

$$\text{BeforeEE}[Ex, Ey] :- \text{CausalRelation}[Ex, Ey]. \quad (1)$$

If  $Ex$  is the cause of  $Ey$ , then the event  $Ex$  temporally precedes the event  $Ey$ .

In fig. 7 an example of an annotation of `MentalByPhysicalEvent` is given.

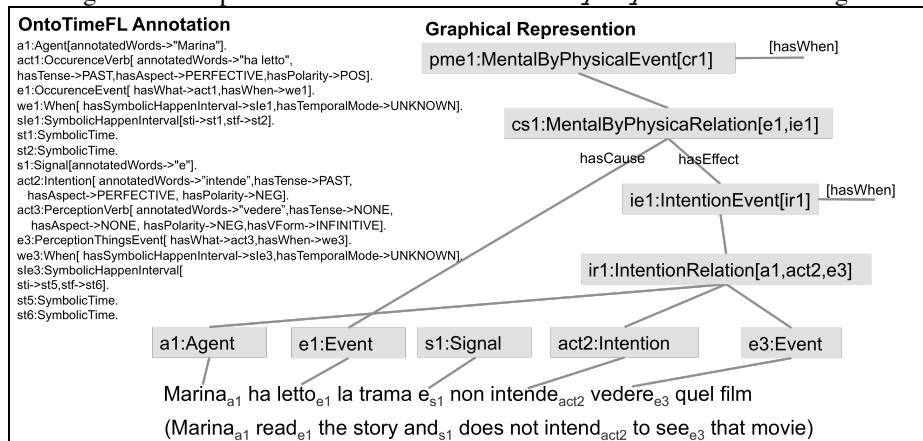


Fig. 7: Example of a `MentalByPhysicalEvent`.

Fig. 8 shows the graphical annotation of a `PhysicalByMentalEvent` event compounded by mental events.





```

Id2:CausalRelation[A,B], C:Event,
Id3:BeforeEE[A,C], newId(Id1, Id2, Id3).

```

The predicate `newId(Id1, Id2, Id3)` generates a new id `Id1` depending on `Id2` and `Id3`.

In the axiom *Left Monotonicity*, the condition `BeforeEE[C,B]` has been included, because the event `C` is an event causing, and therefore to be added, it must precede `B`, otherwise it generates a contradiction.

For the axioms of *Weakening* and *Strengthening*, the meta-predicate `demo(A,B)` (implements the relation "B is deducible from A" [BOC]) has been defined, which was implemented as a variant of the meta-interpreter [BAR].

The axioms for causal relationships shown above have been defined for the class of causal events and can be applied to all subclasses of that class.

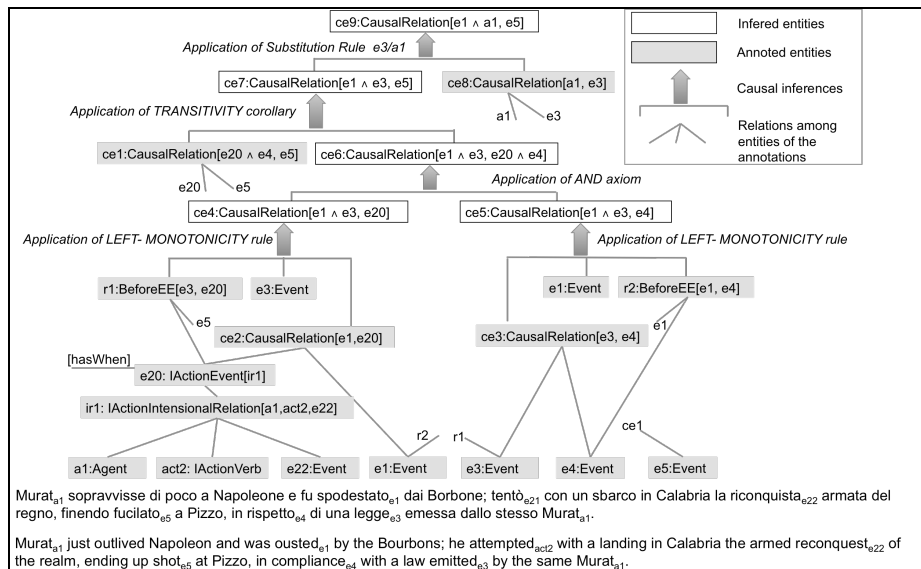
In addition, the following corollaries, demonstrable from the axioms previously provided, have been defined:

```

Id1:CausalRelation[A,C] :- Transitivity
  Id2:CausalRelation[A,B], Id3:CausalRelation[B,C],
  newId(Id1, Id2, Id3).
Id1:CausalRelation[A ∧ B,C] :- Substitution
  Id2:CausalRelation[A∧D,C], Id3:CausalRelation[B,D],
  newId(Id1, Id2, Id3).

```

The axioms, concerning the causal relationships described above, have been defined to be applied to the class of causal events and all subclasses of that class.



**Fig. 9:** An example of simulation of causal reasoning after an OntoTimeFL annotation

Using the formalism OntoTimeFL, an example of a text annotation, (from Wikipedia [http://it.wikipedia.org/wiki/Storia\\_di\\_Napoli](http://it.wikipedia.org/wiki/Storia_di_Napoli)) on the history of Gioacchino

Murat is shown. Causal reasoner, implemented in Flora2, starting from the axioms presented above, has been used to generate the simulation of the reasoning showed in fig. 5. In Fig. 5 entities (events and relations) annotated by the user and those inferred by the reasoner are shown. Among the relations inferred there is `ce9:CauslaRelation[e1^a1, e5]` that claims the shooting of Murat (e5) was caused because of his (a1) having emitted a law (e3) and was ousted (e1).

Through reasoning then, one can build a chain of causes that constitutes the “Why” of events of a narrative.

## Conclusion

In this work, a formalism for the reasoning of complex events has been presented. Attention on three types of events has been focused on: narrative, intentional, and causal. Starting from these three events, the formalism proposed (OntoTimeFL) permits for various compositions of a complex event which have a similar structure to an simple event (the descriptors When, What, etc). The only difference between simple events and complex events consists in the fact that the latter have attributes that emerge from the instances of the components. In this paper, the possibility of associating some axiomatics to classes of complex events has been shown. In this work, one detailed example of the causal reasoning of complex events about a cultural history has been provided in order to demonstrate this possibility.

## References

- [BAR] R. Barták, P. Stepánek, *Extendible Meta-Interpreters*, in: Journal KYBERNETIKA, Volume 33, Number 3, pages 291-310. 1977
- [BOC] A. Bochman, *On disjunctive causal inference and indeterminism*. In: G. Brewka and P. Peppas, Editors, *Proc. Workshop on Nonmonotonic Reasoning, Action and Change (NRAC'03)* (2003), pp. 45–50.
- [BUI] Buitelaar, P., P. Cimiano, P. Haase, and M. Sintek (2009). Towards linguistically grounded ontologies. In *Proceedings of the 6th Annual European Semantic Web Conference (ESWC)*.
- [CAS] T. Caselli, *It-TimeML: TimeML Annotation Scheme for Italian Version 1.3*. Technical Report, 2010.
- [COR] TimeML Corpora. <http://www.timeml.org/site/timebank/timebank.html>
- [COU] Cousins S. B., Shoham Y., *Logics of mental attitudes in AI. A very preliminary survey*. Foundations of Knowledge Representation and Reasoning, LNCS, 1994, Volume 810/1994, 296-309, DOI: 10.1007/3-540-58107-3\_17
- [FLO] G. Yang, M. Kifer, H. Wan, C. Zhao, FLORA-2: An Object-Oriented Knowledge Base Language, <http://flora.sourceforge.net/>, 2008
- [LMF] Lexical Markup Framework (LMF). ISO code number for LMF is ISO-24613:2008. <http://www.lexicalmarkupframework.org/>
- [MEL] F. Mele, A. Sorgente, G. Vettigli, *Designing and Building Multimedia Cultural Stories Using Concepts of Film Theories and Logic Programming*. In *Proceedings of Cognitive and Metacognitive Educational Systems (MCES) symposium*. pag 57-63. ISSN 1613-0073 Vol-598. 2010

- [PUS08] J. Pustejovsky, K. Lee, H.B. Harry, B. Boguraev, and N. Ide, *Language Resource Management—Semantic Annotation Framework (SemAF)—Part 1: Time and events*, International Organization, 2008.
- [ROB] L. Robaldo, T. Caselli, I. Russo, M. Grella. *From Italian Text to TimeML Document via Dependency Parsing*. CICLing (2) - 177-187. 2011
- [TAY] C. Taylor, *A Formal Logical Analysis of Causal Relations*, University of Sussex, 1993.
- [SAU] R. Sauri, G. Lotus, V. Marc, and P. James, *Annotating Events in English TimeML Annotation Guidelines*, 2009.
- [TOO] M. Tooley, *Time, Tense & Causation*, Clarendon Press – Oxford, 1997
- [WAN] H.S. X. Wang, S. Mamadgi, A.Thekdi, A. Kelliher, *Eventory - An Event Based Media Repository*, International Conference on Semantic Computing, 2007, pp. 95-102.
- [WIN] T. Winkler, A. Artikis, Y. Kompatsiaris, and P. Milonas, eds., *Workshop Recognising and Tracking Events on the Web and in Real Life*, Athens, Greece: 2010.